

YOUR

# COMMODORE

AN ARGUS SPECIALIST PUBLICATION

June 1989 £1.50

# BASIC

## ANALYSER

Basic programs revealed

### REVIEWED

- Oxford Basic
- Sketchpad 128

### GAMES REVIEWED

- Deadenders
- Middle Earth
- 3D Pool
- Denaris
- The Deep

### UNBEATABLE PROGRAMS

- 6510 Assembler
- Line Input
- Help Screen
- Retriever

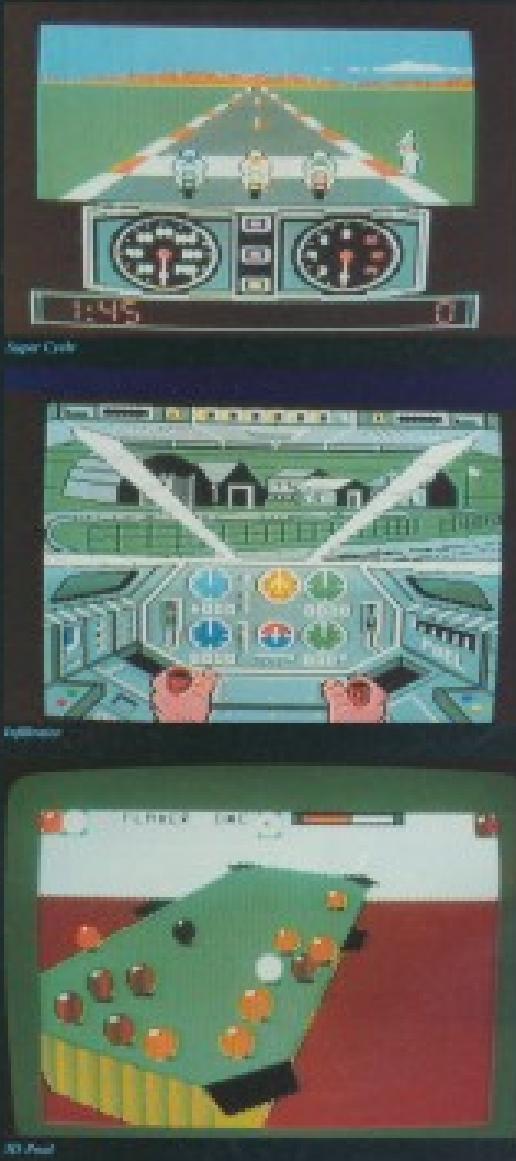






# CONTENTS

VOLUME 5  
NUMBER 9



## FEATURES

• <b>Ideal House</b> Who did it at this famous exhibition	14	• <b>Spy Hunter</b> Popular arcade action	42
• <b>Oxford Basic</b> Expand your C64's Basic with this excellent product	22	• <b>Demaris</b> Superb arcade action	43
• <b>Deadenders</b> No prizes for guessing what happens this is a spoof of!	32	• <b>3D Pool</b> Computer pool as it should really look	44
• <b>War in Middle Earth</b> Adventuring in Tolkein's land	32	• <b>Bargain Bucket</b> The pick of this month's budget games	45
• <b>Super Cycle</b> Motor bike racing at its best	40	• <b>Disk Editing</b> Dive further into your disk-drive	50
• <b>The Deep</b> Under water arcade action	40	• <b>Epson SQ-2500</b> We conclude our look at 24-pin printers	72
• <b>Grand Prix Circuit</b> Try your hand at Formula 1 racing	41	• <b>Sketch Pad 128</b> Does drawing in 3D columns really work?	74
• <b>Infiltrator</b> A second chance to fly with John 'Bull' McGillicutty	42		

## REGULARS

• <b>Data Statements</b> More news from the world of Commodore	6	• <b>Extending Basic</b> Add your own commands to your C64	66
• <b>Letters</b> Your chance to let us know what you think	10	• <b>Tech Troubles</b> Your technical queries answered	70
• <b>1st Steps</b> Help for new programmers	12	• <b>Routine Programming</b> Another handy subroutine to add to your growing library	76
• <b>Flow of Ideas</b> Write your own directory reader	29	• <b>Software for Sale</b> If you don't want to type it - Buy it!	78
• <b>Listings</b> This month's programs	55		

## PROGRAMS

• <b>Returner</b> Call up your disk menu with ease	18	• <b>Help Screens</b> Put your C16/4 HELP key to good use	38
• <b>Line Input</b> Improve on Basic's INPUT command	25	• <b>ASM Assembler</b> A powerful assembler for C64 owners	47
• <b>Program Analysis</b> Find out just what's happening inside your C64 programs	34		

**Editor:** Bruce Cooke  
**Deputy Editor:**  
**Paul Whelton**  
**Technical Editor:** Paul Ross  
**Group Editor:**  
**Mark Webb**  
**Advertisement Manager:**  
**Paul Karamouzis**  
**Add-Copy Control:**  
**Rush Taylor**  
**Artist:** Alan Rutherford  
**Designer:** Neil Stevenson  
**Origination:** Abury  
**Printers:** Chase Webb

**Your Commodore magazine**  
Your Commodore magazine  
appearing on the first Friday of  
each month. Argus Specialist  
Publications Limited, Bishopsgate  
and Abury Publishing Group.

**Commodore Argus Books:**  
Boundary Way House,  
Hempton Road, Tipton,  
West Midlands DY1 1JZ.  
Telephone: 0422-866221.  
Subscription rates upon  
application to Your Commodore  
Subscriptions Department,  
Argus Ltd., 3 River Park Estate,  
Bishopton, Renfrewshire PA4 1HA.  
U.S.A. Subscription Agent: Win  
Dell Worldwide Publications,  
1000 West 20th Street,  
Tucson, AZ 85701 U.S.A.

## ARGUS PRESS GROUP

The content of this publication  
including all articles, designs,  
drawings and programs and all  
copyright and other intellectual  
property rights therein belong  
to Argus Specialist Publications  
Limited. Any reproduction  
thereof by any means (including  
but not limited to photocopying  
and other electronic storage  
and retrieval systems) without  
prior written consent of  
the publishers is illegal and  
unauthorised. Any representation  
in respect of the price quoted  
in this publication is given  
by the publishers only and  
not necessarily those of the  
magazines. While every effort is  
made to thoroughly check programs  
published for errors we  
cannot be held responsible for  
any errors that do occur.

ISBN 0 85039 8277



## Free The Spirit

If there's one company above all the others that's trying to make your C128 more than an Amiga, it's Free Spirit Software. This American company seems determined to produce good, high-quality graphics packages for the C128. Their latest offering is Painter Master 2.0, with which you can create posters ranging in size up to 25 pages. You need 80 col display and a C128D or C128 with the 64K video RAM upgrade. You can import Basic 8, Spectrum 128 or Spectrum 16K files. There is also an option for reducing, thereby enabling you to produce clip art.

**To contact:** Free Spirit Software, PO Box 124, 28 Rocke St, Astoria, PA 19301 Tel 215 681 5488.

## Spirit of Adventure

**G**ood news for Adventure fans. After much public pressure, Innovative Software has re-released the classic Graphic Adventure Creator. The new release will allow 'Stand alone disk accessed' adventures to be produced. This will mean that you can now create adventures of increased size and complexity, restricted only by disk size. The release date for this exciting product is April 1989. It will retail at £29.95, but existing GAC holders may upgrade for a mere £14.95.

**To contact:** Innovative Software Ltd, Zephyr One, Calleva Park, Aldershot, Hampshire, GU1 4QW. Tel 0325 772988.

# Data Statements

**Chris Payne,**  
marketing manager  
at Database, with  
the A128 range



## Stamp Press!

**D**atabase Software have just acquired the rights to two well established DTP packages for the C64 and C128. Both titles originate from AMS. The titles are Stamp Press, and Stamp Press with mouse, costing £39.99 and £49.99 respectively. Pre-processor software is a little thin on the ground for the 64/128, so we will do Database.

**To contact:** Database Software, Europa House, Addington Park, Aldershot, Hampshire, GU1 4QW. Tel 0325 873888.

## Viva Geos

**T**here is no quicker or easier way to present data than with a graph or chart, and GeoChart is a simple and inexpensive way to produce good-looking graphs and charts... etc... etc... as with the blurb goes. Besides Software have added yet another program to their ever growing army of Geos utilities. GeoChart is ideal for anyone that needs professional-looking documents and charts, but can't spare the time needed to plan and draw graphic illustrations.

Like all the Geos appli-

options. GeoChart is in the "what you see is what you get" format. That is, pull down menus, windows and icons. GeoChart is for the C64 and C128 owners.

**Powerline Berkeley Software**, 2129 Shattuck Avenue, Berkeley, California 94704. Tel: 415 549 0881.

**GEOCHART**  
A Geographic Information System  
by Powerline Berkeley Software

## Hubbabubbabubba

Electronic Arts have enhanced their popular game from Bullring, *Populous*, by adding a new sound track from Rob Harbeck.

Rob, who left the UK in '88 to work in sunny California, has since been working on a major project. The work involves creating a demo package that will allow real soundtracks to be produced. The driver, we are told, will be available soon and even again for any product that requires a soundtrack.

**Powerline**: Electronic Arts, Complex Business Centre, 11-19 Station Road, Langley, Berkshire, RG1 8TN. Tel: 0753 494422.



Rob Harbeck - author in exile

## Watch Out

**M**icropress are about to launch an attack on your wrists with *Navy Seal*, a \$4 game packed with action. You are a member of the elite US Navy force, the semi-sub-based commandos, who are experts in sabotage, demolition, reconnaissance and infiltration. So that you don't

miss the beat, a *Navy Seal* Digital Watch (right) will be given away with every copy purchased. The release date should be in early April, at an expected cost of £12.99 or disk.

**Powerline**: Micropress, 2 Market place, Tisbury, Wiltshire, SP2 8DA. Tel: 0861 342121.



## Make a Stand

**H**omecomputer Furniture never gets a mention. Well, here goes. From MIDS of Hemel Hempstead comes the MIDS 750 printer stand. Of all the peripheral printers, printers seem to cause the most positional headaches, but this new stand could solve all your problems (although the £204 price tag may put you off). It is, and I quote "uniquely designed to achieve maximum efficiency, and part of a comprehensive modular desk

system which measures 750mm wide, 710mm high and 750mm deep, is an 18 inch depth on the top, making the unit suitable for both top and bottom feed printers". The other good point is the electronic cable management system incorporated into the unit - no more wires and cables need get in the way.

**Powerline**: MIDS Industries (UK) Ltd, Factory 1, Elstree road, Hemel Hempstead, Herts HP2 0QZ. Tel: 0492 221265.



## Change At Waterless

Perseus of Restricted Growth everywhere, here's your chance to get your own back! The latest offering from ESI, available via US Gold, allows you to assume the persona of 'Sharks' Bonaparte and change the course of history. Called *Battles of Napoleon* - a contraction set, the programme allows you either fight authentic Napoleonic wars, or simply create your own. We are assured that these battles are realistically recreated in every detail, with varying charges, bombardments, skirmishes and formations. Priced at £14.99 for the C64, the game offers good value.

Touchline: U.S. Gold, 2/3, Halford Way, Halford, Buntingford, Herts, UK. Tel 021 556 2388.

## Special Relationship

**N**ews has reached us of a bilateral deal between Krayola and Action Ware,

the omission of which should prove beneficial to all you game players. The deal means that Krayola will market AW software here in the U.K., and AW will market Krayola over there. The first product off the press was *Prison*, which was released on March 16th for the Amstrad ST and Amiga.

## The Hammy Hamsters

Fame and fortune can make people do weird and wonderful things. Howard Hughes became a recluse, a certain Mr. Khan bought a sugar factory, and Mr. Wyman managed every man's dream. Now it's the turn of the Darling family. Not content with purchasing both possible budget software, they have recently broken out into song. To help promote their latest album, *Rock Star*, they've formed a group which, for reasons best known to themselves, is called The Hamsters. Judging by the picture, I know which Hamster Ed like as a pet!

Touchline: Code Masters



Software Co. Ltd, Lower Farm House, Stoneybridge, Southern, Wiltshire, SP13 0DA. Tel 0980 674712.

## Hit The Jackpot At The Commodore Show!

### Amiga Airlines

Based on Commodore's new 'Amiga Air Miles Pack' will be carrying free air miles along with their machines. The package offers an Amiga 500, a TV modulator, three new games (*Roger Rabbit, Nibaldo and Star Fox*), congressional point packages and 300 free air miles, all for £499.99. The offer will give Commodore customers the chance to book return flights to a number of European cities, including Paris and Amsterdam. All flight bookings will be valid until March 1990, and can be easily redeemed via the special Commodore Air Miles hotline, as well as local travel agents.

Touchline: Commodore Business Machines (UK) Ltd, Commodore House, The Shrubland, Gardner Road, Maidenhead, Berks SL6 7LA. Tel (0628) 73455.

Commodore are giving away £300 worth of computer equipment to the Amiga user with the most innovative and unusual use for his machine. A nationwide search has been launched to discover as many weird and wonderful applications for the Amiga as possible, and entries will be judged by a panel of experts at the Commodore Computer Show, which opens at the Novotel, Bournemouth on June 2.

The show itself has been revamped by Database Exhibitions, and will now include seminars on making music, graphics, Commodore magazines (including, of course, PC) and game writing. Best 'The Author' competitions are also planned, in which writers will be able to take on "named" authors and try to beat them at their own game. But one of the big highlights of the show will undoubtedly be the innovation competition.

Commodore has already discovered a wide variety of uses for the Amiga, from bird-watching to betting on horses, and has even heard of one user who writes out menus for rapids on his machine! If you think you can match these innovative applications, all you need to do is write a brief description - no more than 500 words - and send it in, along with any necessary support material on disk, to:

The Editor,  
Your Commodore,  
Amiga House,  
Boundary Way,  
Brentwood, Essex, CM1 3LT.

Entries should reach us before May 31st.



# Mailbag

## Your chance to air your views in Your Commodore

I have just received a Commodore PET 2032 Computer from a friend. With it was a 3250 Disk Drive and 4828P Printer. I was wondering if you or any of your readers had any information on software availability. Please send any information to me, thank you.  
R.M. Crook, 29 George Road, Bromley Cross, Redhill, RH2 8AU.

ribbon? I feel that it's a great waste to throw them away.

Could one possibly revalue them with ink in some way? This, which ink should be used, or can I send them somewhere to be done?

J. Gibbons, Luton, Beds

### Our reply

The answer, John, is yes. Private ribbon can be revalued. If you get a copy of Micro Computer Mart, you will find a free advertising service every week.

### Dear VC

With three CDU disks, I have had some loading problems. The fault lies not with yourselves, but rather with the magazines, when they stuck magazine on top of each other. I have found that "Validating" the disk beforehand seems to cure this problem.

I don't know what proportion of failures are returned, but this tip may help. J.F. Pashlock, Wood, Tewkesbury, Gloucestershire, GL20 7AS.

### Our reply

Thanks for this helpful tip-up. You do get some returned disks, and many of these are cured by this simple method.

Unfortunately, newsgroups and bookshops also receive few delicate disks are.

I was recently given a PET 2032, a 3250 Disk Drive and a 4828P Printer together with four good software packages complete with manuals and guides, where necessary. I only want to use the PET for wordprocessing, and so I am sending out an appeal. Is there anyone out there that has a program for the 2032 that they don't want? If so, please contact me at my address. Thank you.  
J. Simpson, 4 Goldthornes Avenue, Dunbar, East Lothian.

Come on now everyone, let's do it! There are pretty ancient machines, but there must be someone out there who can help!

### Dear VC

I have anything that can be done with old printer

### Dear VC

I have been comparing ever since the Commodore 64 was first produced. In this time I have bought various computer magazines, mostly multi-format, and I've noticed various discrepancies between reviews of the same game. For instance, Temis received 58 out of 100 from one, but 97.5 from another.

The trouble with two completely different results is which one do I believe? Do I take the advice of magazine A and stay clear of it, or do I take B's advice and buy my TV so that I can buy it?

The best advice for anyone is to try and get into the reviewer's mind. Some reviews set their hopes too high, and so if a game doesn't come up to these expectations, they give it a poor write-up.

A recent review of Thunderbird is a good example of this. One reviewer of this game said - "The graphics at this point, Level 1 stage 1, were very disappointing - the buildings are just stacked squares".

Personally, I think the graphics are very good. Okay, the first stage on each level is blocky, but you can't really expect hi-res graphics can you? It would be much too slow. Anyway, you've always got stages 2 and 3 to each level to make up for it.

Another annoying item is "Missing". You know the kind of thing - "The Q button doesn't work, you have to use the F7 key". A great example of this can be found in the same review of Thunderbird. The reviewer said "Unfortunately, it is hard to find any difference between cassette and air-to-

ground rockets." There are two possible reasons for this:

(A) His/Her cassette deck/disk drive was faulty or (B) their copy of the game was faulty. These are the only possibilities, as my copy of the game is the opposite.

In conclusion then, what do you do if two reviews differ? The first option is to see if you can have a demonstration of the program, therefore allowing you to make up your own mind. The second option is to simply take a chance.  
N. Kingsley, Chichester, West Sussex

### Our reply

Thanks for such a long and detailed letter. However, I think you're missing the point (as do a lot of people). Reviews of software do not differ from any other form of review, be it film, video, records, food, clothes etc.

The whole point here is that - the comments passed are the comments of that particular person, NOT the comments of a 'professional critic'. For example, I personally have never liked Elton, but for 1.3 million other users it is the best program ever.

Likewise, for me Infocom are the only people that produce decent adventures, but over thousands of people would not agree. Basically, one must remember that a reviewer's comments are his own personal comments. A review only really tells you what the objective is and how the game plays. There is only one person that can say whether it is good or bad, and that is you.

# MIXING Business

## WITH L, E, I, S, U, R, E

### WILL BE A REAL EDUCATION

**C Commodore computer show**

Britain's brightest event for Commodore computer users is back! And there's more to see than ever before.

This show has three main themes covering some of the major uses to which Commodore machines are put. There are over 80 firms comprising who will be exhibiting their latest products, while visitors can find almost everything that's new in the Commodore world will be on show!

#### Business

Many companies will be demonstrating their latest software and hardware, especially designed to increase the business potential of Commodore computers.

As well as products for the C64 and Amiga series, you'll also be able to try out applications for the promising Commodore PC-compatible range.

And you'll also be able to attend seminars covering all aspects of using Commodore machines in your business.

#### Leisure

The C64 and Amiga computers are the most powerful and flexible means for producing fun action arcade quality games. The range of new software on show

— Novotel Exhibition Complex,  
Hammersmith, London W6

Friday to Sunday  
June 2 to 4

Open 10am-6pm Friday & Saturday; 11am-5pm Sunday

will demonstrate how these machines' power is continually being stretched, producing faster and even more addictive games with superb graphics.

If you're a keen game player, you'll find there's as much as you can offer at the show you've guaranteed a real treat!

#### Education

Commodore machines are now used as educational tools all over the country. With the development of BBC Basic on the Amiga, and the advent of Desktop Video (embedding TV pictures with text and graphics), the range of educational applications is enormous.

At the show you'll see how the latest software

packages are making real breakthroughs in the educational sector, and be able to try them out for yourself.

#### Special Events

In addition to special events and presentations, you'll be able to view some of your favorite video cassettes, and visitors get a chance to talk with them about how they use music in their work.

So for a great day out, whether you want to see what the future holds for Commodore computers, to buy the latest software or to get advice on specific applications, the Commodore show is the place to go. And if you send in the coupon today, we'll knock £1 off the price of each ticket!

• For the first time we are offering a family ticket for just £11 allowing entry for two adults and two children — saving up to £7 off the usual entry price!

#### Please Tick These

**By Cheque/Postal Order:** Please make payable to Commodore Computer Show Ltd, Novotel Exhibition Complex, Hammersmith, London W6 7AA.  
**By Fax:** 081 794 704 288, 29 31, 73, 76  
 Car parking facilities available at the Novotel.

**Advanced Ticket Order** **C Commodore computer show**

Novotel Exhibition Complex,  
Hammersmith, London W6 7AA

- Cheque payable to Commodore Exhibitions  
 Please debit my Access/Visa card no. \_\_\_\_\_

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Expiry date: \_\_\_\_\_

Please tick here:  
 Adult tickets at £14 (total £11) \_\_\_\_\_  
 Under 16 tickets at £12.50 (total £10) \_\_\_\_\_  
 Family ticket at £11 (total £7) \_\_\_\_\_  
 Total £ \_\_\_\_\_

Signature: \_\_\_\_\_

Address: \_\_\_\_\_  
 Advanced ticket orders  
 22 Gloucester Place  
 22-26 Gloucester Place  
 London W1A 4JL

Advanced ticket orders  
 for collection by  
 Wednesday, May 31

Name: \_\_\_\_\_

Address: \_\_\_\_\_

Postcode: \_\_\_\_\_

**POSTAGE PAYABLE** Ring Shows Booking 081 794 2881  
 Mon-Fri, 9am-5pm, 9am-1pm Sat & Bank Holidays  
 Novotel Exhibition Complex, 22 Gloucester Place, W1A 4JL  
 Please quote reference number and telephone no. **AESD**



*How to make sense of Listings* By Norman Doyle

Typing listings in can be much more of a trauma than it need be. At Your Computer, we are aware that sometimes the process seems difficult or even impossible. Our Listings with their Syntax Checker and special codings can ease the way, but sometimes more support is required.

Many programs are listed in several parts and each part uses the same line numbers. How is this possible and how do you set about typing them in?

First of all it should be explained that a set of these programs forms a unit which operates in unison. The way to use them is to type in the first program and run it. Now the computer should be cleared by pressing NEW or even by switching the computer off and on.

Proceeding to the second listing, the program is again typed in and saved. If you're storing the programs on tape, you should ensure that this program is saved immediately after the first one. For disk users, it doesn't matter where it's saved as long as it's

on the same disk as the first program. After checking that all has been saved safely, the machine is again refreshed using NEW or the power switch.

Any subsequent programs should be treated in the same way as the second listing and the result will be a string of programs. One word of caution which should be heeded is that the programs should be saved under the name given in Your Computer because each program occasionally uses one of these names to load in the next program in the string.

### Suite Confusion

One question which we're often asked is why these multipart programs are needed. The answer is that they are a memory-saving or time-saving measure.

Many programs use redefined characters, machine code patches and updates. The information for this has to be packed into positions because they rarely occupy memory locations which BASIC uses. This difficulty can be overcome in either of two ways.

The data can be packed into positions each time the program is loaded and forms an integral part of the program. This has the dual disadvantage of the time the program takes to move the data in but, more importantly, reduces the amount of space for the actual program itself.

Another way of achieving the same result is to use a series of set-up programs before the main program is loaded. These set-ups are loaded and run in turn and each one packs information into memory where it cannot be touched by any subsequent loading actions. Consequently, once the program has done its job it is no longer needed and the program can even be removed by typing in NEW without raising the effect of the main program when it eventually loads.

One thing that you cannot do is to switch the computer off and on again! This action totally clears the memory, including the data which has been packed in.

When the first program has done its job, the next program is loaded normally and run. Now there are two

blocks of data located away safely. This continues until the final program is loaded. This is the main program which holds the key to extracting and using the data which the previous programs positioned. The fact that each of the programs may have contained the same line numbers is totally irrelevant because it's what each program does and not how it does it. It's a little like receiving a birthday card and an electricity bill - they both carry the same address but the effect they have on the mind is totally different!

### Gift Boxes

Although this system solves the problem of memory space, it does use time and a long program can seem to take forever to pack the data into position. One way to overcome this is to create a program which not only packs the data into position, but also saves the pure data block afterwards.

These two-stage programs, or basic loaders to give them their correct name, are quite common and need treating separately. The process can be

quite complicated so we'll take the case of a single program first.

As usual, the program is typed in and saved before running it. Next a new tape or disk should be placed in the storage device. Now when the program runs it will save the data back onto the new tape but in a different form. The new program will be loaded directly into memory without the need for a basic program to pack it there. These programs usually have to be started with a SYS command and, once cleared, the program that was originally typed in can be forgotten and erased.

The Basic program is rather like a gift box which becomes useless once the contents have been removed.

A program of this kind which is separated into parts is immediately recognizable because the master program is the first in the chain. If this is in Basic it should be typed in and saved. If it is a Basic loader it must be treated differently.

### Spot The Loader

A Basic loader can be recognised by

the reuse of DATA statements which must be typed in. Another indicator is thelessness of the actual program, excluding the DATA statements, which often only contains a FOR..NEXT loop separated by complex POKE commands.

If the main program proves to be a Basic loader it should be typed in and saved but when it is run a separate disk or tape should be already in the tape device.

After dealing with the main program, any subsequent parts should be typed in and the version saved to the final tape or disk. On running, the resultant program should be saved as the second tape immediately after the preceding part.

Finally, the rule to remember is that any programs which have duplicated line numbers cannot reside in memory at the same time. They should be typed in and saved separately according to the instructions given in the relevant User Commands article or the REM statements in the listings themselves. Stick to this rule and you'll never go wrong... well, rarely!

## TELETEXT

A world of information  
at your fingertips



Teletext on your BBC or 128 brings you the very latest information from America. With a Teletext TV, you can print pages like today's TV, view a colour horizon or computer. The ability to receive Teletext data from your own programme provides endless possibilities. Comtex and Oracle provide hundreds of pages of news, sport, prices, weather and general reports, over foreign holidays plus much much more.

The Microtext Teletext Unit costs £199.95, ready for use, just connect it to the Teletext and play in or out of the Adapter. Also may be connected to the VIDEO OUT socket of a video recorder.

The Microtext Adapter is only £99.95. Adapter and Teletext cost £299.95 including VAT and p&p.

#### NEW Upgrades

The Upgrades allow your old Microtext Adapter to be converted to the Adapter and comes complete with Atrage software for only £24.95.



**MICROTEXT**

Dept. PD 7, Farington Close, Hornbeam, Hants PO18 0PR  
Telephone: (0705) 586504

#### OFFICIAL COMMODORE/AMIGA DEALER

ITEM	PRICE
COMMODORE 64	£199.95
COMMODORE 64 WITH 128K RAM	£219.95
COMMODORE 64 WITH 128K RAM & 1.44MHD	£239.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE	£259.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 1.44MHD	£279.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY	£299.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD	£319.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD	£339.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION	£359.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD	£379.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD	£399.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD	£419.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£439.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD	£459.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£479.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£499.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£519.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£539.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£559.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£579.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£599.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£619.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£639.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£659.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£679.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£699.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£719.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£739.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£759.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£779.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£799.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£819.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£839.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£859.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£879.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£899.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£919.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£939.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£959.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£979.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£999.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1019.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1039.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1059.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1079.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1099.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1119.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1139.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1159.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1179.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1199.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1219.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1239.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1259.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1279.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1299.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1319.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1339.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1359.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1379.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1399.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1419.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1439.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1459.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1479.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1499.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1519.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1539.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1559.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1579.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1599.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1619.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1639.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1659.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1679.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1699.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1719.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.44MHD & 1.2MHD & 512K EXPANSION & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD & 1.44MHD & 1.2MHD	£1739.95
COMMODORE 64 WITH 128K RAM & 3.5IN DISK DRIVE & 3.5IN FLOPPY & 1.	

# The Case of the Electronic Home

**H**imes, I've got a letter here from a mysterious organisation asking us to investigate the future of the domestic house, but I don't know how to go about the job." "Well Watson, you can always pop down to the furniture exhibition at Earls Court and see what's new." "That's Ideal Homes?" (Oh God...)

So it was that our two hump-backed detectives went off in search of clues as to how long it would be before everything in the house could be operated at the flick of a switch.

When they arrived at the show, though, they were in for something of a surprise. Whereas even a couple of years before, every machine was being heavily promoted as the all-engulfing, all-dancing model with more lights and buttons than you knew what to do with, this year's displays were much more minimalist in design. It was as if it was assumed that the machines could do everything that was asked of it, and it was simply left at that.

"There are two distinct problems here," said Holmes. "To start with you will notice that just about every machine, from cameras to washing machines, has chips in it, which in turn gives the end user access to a plethora of functions. But although the owners may appreciate having all these extra facilities, he doesn't usually use many of them. After all, just about every single computer disc player lets you programme the tracks in any order that you want. But who in their right mind is going to mess about with the order of the movements in a Mozart symphony?"

The second problem is that nothing is compatible unless you wish to just use manufacturers' So the customer ends up with separate remote control handsets for his television, video and Hi-Fi. None of them will run on his dishwasher, and he can only use them over a limited range. What is needed is some all-powerful control system.

*Leaving aside his relentless search for the dreaded Moriarty, Mr Sherlock Holmes and his old friend Dr Watson visit the Ideal Homes exhibition and discover that 221B Baker Street is not all that it should be...*

By Gordon Handlett



"I notice that all the display houses have a 'study' contained," Watson. "The idea of people working more and more from home persists, even though there is little evidence to suggest that it is actually happening. It seems that computers are still regarded as toys unless they possess those three magical letters - IBM. Moderns still have not caught on as they have done in

America. Instead, no home should be without its own personal fax machine."

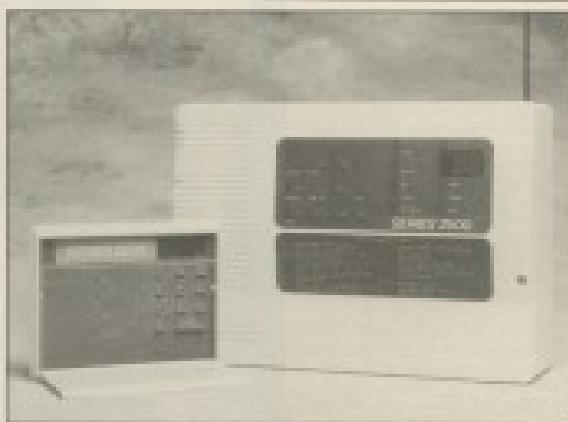
"If we are not very careful Watson, you and I could soon be out of a job. The one major area of expansion is the electronic. That is in security systems. I see that Modern Alarms are offering a wireless programmable system - easy to install, no messing up the decoration, and it can be taken



with you when you move. And every type of alarm imaginable is here—magnetic contacts, heat, radar, sound, vibration, breaking glass, infra-red beams and even current sensing. There are even personal alarm buttons."

"How is the enterprising burglar supposed to get past that little lot, and if there are no criminals than what are we to do? Ah, here is something that would fool even me. A machine that emits artificial barking voices. Remember that case of the dog that didn't bark in the night? This device would have totally messed up that story."

"I can find no trace of robotics here Holmes. It looks as if we will have to put up with Mrs. Bradman's housekeeping for at least another year. This is one field that has definitely not progressed as quickly as everybody anticipated, although I still hear rumours that two of the biggest fast food chains are hunting to be the first



to get their customers served by automation.'

'At last, Watson, I think that we have found what we were looking for. A system that will control every domestic appliance in the house and which can be operated either from the central processing unit, from a handset, more importantly, via a telephone link. Just think of it. Being able to phone home if we are out on a case and instruct the radio to record Crimewatch and Police 2.'

'I see that the system - Crodcast from Croda - works by means of a series of transmitters placed in every appliance. It then communicates with the central processing unit using existing mains-circuitry. This guy has a large display panel as well, so that you keep addresses and diary dates as well as being able to check up on share prices and train times. That would have been useful in the Baskerville case.'

'We would be able to control the temperature in every room in the house, varying it as and when we want it. Lighting could be similarly controlled, ranging from individual lamps upwards. Imagine being able to turn out the kitchen light from your bed when you had forgotten to do so. All your alarm-related security devices could be checked easily, although it would lessen the impact of my detection powers if I knew beforehand exactly who had turned up at my front door.'

'Turning on the cooker, doing the washing in the middle of the night and arranging for seconds of my beloved radio music to start playing as soon as I walked through the door. And all available in the next couple of years for about a thousand pounds.'

'This has got to be the way forward, Mr Watson! This is where the future of the electronic house lies. It doesn't matter what make of appliances you've got - they can all be controlled from the one system. I would guess that we will be hearing of many more such control systems in the next five years, each offering more and more features and with the price coming down all the time.'

'There's only one thing left for us to investigate now, Watson, and that is to find out what is on the other side of that pillow door? What's so special about the pillow door, Holmes? It's a lesson entry my dear Watson! (More cough, Watson, no more phlegm - it's through Ed).



# Quality. Designed from the inside out.

The Business Series



Value



24



Simplicity



Fast



Paper Feeding



Speed



Compatibility



Support

For state-of-the-art printing, the mid-range LC24-B gives you value for money as well as the top-quality quality of a 24-pin printer.

**L** The LC24-B is compatible with almost every computer's serial interface; the printer is also suitable most industry standards with over 2 standard formats that we think should always be included. For connection over longer distances, there is even the option of a low-cost serial converter.

**L** Putting the LC24-B to work couldn't be simpler. At the mere touch of a button on the front control panel you can choose any one of eight excellent letter quality fonts, determine your

print, As well as all these features, it offers two distinctive unique highlights for writing business.

**L** Printed documents in one second and that's just the start. The LC24-B will also perform word perfectly in single pass producing exceptional letter quality for polite correspondence and important documents. And while you print, the LC24-B will free up your PC for games with other things.

**L** No other 24-pin printer in this price range can match it. Features like its generous font buffer, expandable to 192K, built-in pitch control and (optional) continuous paper feeding facility are all included as standard. Another impressive feature you won't find in an optional extra is the LC24-B's remarkable range of print variations. All items come with the option of shadow, outline and even shadowed outline to meet all requirements.

**L** BusinessStar printers have a guarantee of superb reliability together with a comprehensive back-up service for peace of mind. Should you need any help at any time, you can rely on our national three-level support system which operates throughout the dealer network and is backed by the three distribution centres in the country.

**THE**  
**star**  
COMPUTER PRINTERS

Star Microsonic U.K. Ltd.,  
Cavers Mound, Oldbridge Road, Andover,  
Hampshire SO10 2RS. Telephone 0303 640 1000.

A Division of Star Microsonic Co., Ltd., Japan.

Please send me all the information I'll need. Alternatively, please call Andover 0303 640 1020.

Name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

Postcode \_\_\_\_\_

Telephone \_\_\_\_\_

STARV7

All trademarks acknowledged.



Public sector customers may purchase or pre-order items from STARV7. Contact 0303 640 1020.

# Returner

*Return to the menu at any time with this ingenious program*

The problem of keeping records of what's on your disk was addressed in Commodore Disc User Vol 1, No 2 (Jan/Feb '88) by Mooy Makler. This useful utility allowed you to load any program direct from the menu, but to return to MENU you had to add the load statement to each program on the file, which is easy enough to do, but usually means posing check questions in and having to end the running up of the program. You could also do it by pressing Run Stop and reloading "Menu", but that's rather a waste of time.

The program Returner allows you to leave a program at any point or time, and return to the menu to select an alternative program without all the problems outlined above.

On loading and executing the "RETURNER" program code, the first job it does is to set up the "RESTORE" key, and automatically load a MENU.

Obviously, for the program to work correctly, you must have a "MENU" program on your disk. We have published quite a few such programs in the past. Alternatively, you could write one of your own. To get the best results, it is advisable to have at least two, if not more, programs on your menu.

Once you have your menu program running, you select which program you want to run. On running the selected program, you can press the "RESTORE" key at any time to return to your MENU program.

## If the returner won't work

There are two reasons why Returner may not work:

1. The program in memory is using the SMI interrupt, or it is intercepting the interrupt vector to normal values. You need to alter your program (if possible) so that it's not using the SMI interrupt.

S.M.I. The SMI interrupt values are located at 792-793 (\$118-\$119).

2. The memory location where "Returner" is located is being used by the program being run. See solution below.

## Program checker

This program will provide you with alternative memory locations to hold the "Returner" program. Load "CHECKER" & run. This will give the following instructions on screen:

File Type in N  
Then Load Menu, Load A File, & Run  
Load Menu, Load Another File, Reset  
Repeat Until all Files Loaded  
Load Checker & Type Y  
Have You Wiped Before?

When running, "Checker" will show a blank screen and processing is complete. This will take approximately three minutes and then show on the screen:

Ready  
Load "Menu".B

Press Return, Run the menu, then load file file, then reset computer by pressing K1-N1STOP & RESTORE or by using a Reset cartridge. Do not switch the computer off & on.

Reload the menu, run the menu program, then reset as before and continue until all files have been loaded. Once all the programs on the disk have been run, reload CHECKER, run, and type in Y to the question.

Output is given to output to printer as well as screen. The result of the program is a list of free memory positions available for insertion of the program Returner. You can select the desired position to place Returner (if no space available it will tell you so, i.e. Returner cannot be accommodated on that disk.)



To change the position of Returner in memory

Load "EDIT".B and run

The screen will ask for the new start address. You can enter it in Hex or Decimal numbers (the Checker program provides location values in decimal).

Once entered, press return.

The screen will then ask you to state which file you want to be automatically loaded when you press the RESTORE key. If not MENU, then alter to your own requirements - press Return.

The screen will then ask what name you wish to give the Returner program. Put your own name in. The screen will then ask "Are You Sure?"

# Returner



- if not, type N, if you are, type Y. N returns you to the beginning of the screen, and you will have to reinput memory location etc. When you type Y, it will pass the newly-loaded program to disk.

## Starting up your computer

When you start up your computer, you normally load and run the program you named and saved above. It will automatically install the RESTORE key function and run the program you asked to be first loaded (see above).

I recommend that it is the menu as pressing the RESTORE key will automatically return you to this program until the computer is switched off.

**Note:** A good place to put the "Returner" program in memory is \$28 (\$324), as it is not normally used when using disk. If it relates to the tape drive too, so you won't be able to use the tape drive.

If you have a multipart game, Returner can be used to restart the game by asking it to automatically load in the first part of the game as

part of its program (see above). The start program must, however, have a basic line number at the beginning e.g. 10 SYS 4880.

## Addition to Returner instructions

It is not recommended to press RESTORE during the use of the disk drive or printer, as the menu may not work properly if you do so.

If you need to press RESTORE while the printer or disk is in use and the menu doesn't work properly, then just press RESTORE again and the menu should work properly.

## Financial Systems Software Ltd

### Technical Books

**Primer Book for the 64.** An insight handbook which covers everything you need to know about assembly language programming, memory management and more. Includes software disk. 300 pages. Only £14.99.

**Compiler Design & Implementation 64-bit.** Learn how to design a program to compile your software. Learn to design a language suited to your specific needs. It will write a corresponding compiler. Appendix A: Selected 64-bit Assembled Books. Software disk. 390 pages. Only £14.99.

**Software & Engineering for the 64.** An introduction to using the 64-bit architecture. Contains sample code, numerous useful memory commands in assembly, basic I/O routines, registers, CPU register distribution, Fourier analysis, matrix calculations. Book is disk. 300 pages. Only £14.99.

### Selected 128 Books

**128 Books Information Books.** Conveniently one of the 128 commands, functions, etc. from assembly language functions. Instructional, diagnostic programs, memory commands, I/O routines, etc. are given. A must for all 128 users. Only £14.99.

**Perils & Pitfalls.** A collection of useful posts and notes for the 128 and the new book 8 software disk. 200 pages. Only £14.99.

**Topics & Tips.** A valuable source of programming hints, including disk pointers, graphics, sound, and much more. Book & software disk. 300 pages. Only £14.99.

**128 Books 2.0.** A very comprehensive guide to the 128. With over 1200 pages to the largest 128 book. £14.99 (softcover) disk £1.99.

**128 BASIC Training Guide.** A full textual guide to programming in BASIC. Disk £14.99. Full reference and documentation. Book & software disk. 200 pages. Only £14.99.

### GEOS Books

#### GEOS Inside & Out

The most complete guide to GEOS using GEOSIDE, GEOS Inside & Out gives the beginner a gentle introduction to controlling GEOSIDE. Later chapters present the user with GEOSIDE & GEOSIDE. One chapter is dedicated to practical uses, other topics include assembly & tips, creating custom windows for your programs, GEOSIDE file formats, etc. Numerous, A costly but comprehensive guide to the GEOS manual. Includes GEOSIDE & Inside & Out books. Version 1.0. 200 pages. Only £14.99.

#### GEOS 2.0 Inside & Out

A guide to using GEOSIDE, and the GEOSIDE 2.0 experimental user. Over 150 topics are included. From setting up documents to GEOSIDE to batch menus, printing files, programming your own serial messages and much more. Programs included & supplied on disk are: Counter, memory dump, reference & prompt, a font editor & converter. Book & software disk. Over 200 pages. £14.99.

### Official 128 Guide

Commodore's official programmers guide to the C128

- indispensable reference guide for the C128
- known bugs included
- The new BASIC 2 - response time commands
- BASIC 2 - utilizing the C128 graphics power
- GEOSIDE & Inside - getting the best out of the C128
- GEOSIDE - getting the best out of the C128
- Graphics - memory management
- Graphics - memory maps
- Input/Output - peripheral control
- Disk - floppy disk interface
- Over 200 pages. £14.99.

### 128 Repair Guide

Problems with your C128, are you an experienced or inexperienced computer owner? Learn how to repair your component?

- Troubleshooting guide by request. Price
- unique test finding on the 128
- indispensable described in colour
- Serial port equipment & troubleshooting
- Parallel port equipment & troubleshooting
- Video card problems & troubleshooting
- Power supply maintenance advice
- Early and late the 64 & repair yourself
- Large, low cost book, full of diagrams. £14.99.
- New Troubleshooting Guide £14.99.

### Anatomy of the 128

Inside your C128 in great detail

- C128 Disassembling the heart of the C128
- Programming Macro & Record
- Programming graphics, modes
- Using the 128 colour chip
- ROM processor and boot ROM
- Assembly language programming
- Full references to C128 listing
- Programming the ports
- Optional disk - contains basic programs £14.99
- Over 200 pages full of information. £14.99.

### Anatomy of the 1571

The essential reference guide for all 1571 users.

- For the Beginner & advanced user
- Fundamentals for disk drive beginners
- Applying disk drive commands
- Creating Utilities & Sequential files
- Using either 1571 or 1570 (disk)
- Basic access commands
- Working with foreign disk formats
- The 1571 OS system
- Putting programs on the 1571 buffer
- The C128 1571 and C1570 software
- Internal disk drive functions
- Many utilities including disk & file copy
- Fully documented 1571 listing
- 1571 Directory layout
- 1571 & 1570 disk drive issues
- Optional disk - contains basic programs £14.99
- Over 200 pages. £14.99.

### 1541 Repair Handbook

The book 1541 users have been waiting for

- The 1541 repair & maintenance handbook
- A complete guide to repairing your disk drive
- Maintenance procedures & intervals
- Disk motor speed adjustment
- Testing for current starvation
- Adjusting the read/write head
- Head cleaning - memory maps
- Memory map - memory dump
- Complete description of the electronics
- Detailed diagrams on most parts
- Detailed disk & controller board programs
- Large A4 format book with 300 pages. £14.99.

### Hackers Book Pack 64

- Perils & Pitfalls
- A lot of useful posts. A guide for programmers
- Program errors, sounds, save/restore, etc. more
- Computer systems, sound, memory, etc. more
- High resolution graphics, keyboard, user port
- BASIC, assembly and machine language
- Over 200 pages packed with information
- Topics & Tips
- A collection of useful & interesting
- Includes information covering -
- Graphics, colour, sound, 3D, more
- Advanced BASIC, more, C128
- Including Bitmapped colors
- Disk management & working, plus much more
- Over 200 pages
- Hackers book 64 including both titles £14.99

### Selected 64 Books

**Anatomy of the C128.** Packed with information on the 128. In 6 parts. Chapters include graphics, sound, memory, mode, BASIC, assembly language and PC/AT settings. 300 pages. £14.99.

**Anatomy of the 1571.** Remove the mystery out of your 1571. Learn with the aid of diagrams & programs how to use them. Use the 1571 to develop, including data to the 1571, directory writer, file manager, file transfer, BASIC, language & file support. Contains commented BASIC listing. Over 200 pages. £14.99.

**Basic for your 64.** Make your 64 work, many practical ideas are given with their software to nearly use possible. Perfect for the novice, some of the projects covered include colour screen, advertising, sequence monitor & memory card and more. Over 200 pages. £14.99.

**Geoside Book 64.** A straight forward book that teaches you how to program, set and change graphics. Create C64 pictures, have character sets, titles & much more. Includes, graphics & memory, I/O, graphics animation, chip control and much memory management & more. Over 200 pages. £14.99.

**The Official 64 Programmers Guide.** A complete basic for the 64, published by Data handle. Topics include assembly language, memory, graphics and more. All parts of the 64 are covered including: Commands, BASIC editor, BASIC loader, BASIC interpreter, memory map, etc.

Optional disk available for selected books only £4.99 per disk.

## NEW - Commodore 64 Graphic Software

### Clip Art Disks

#### 1000 Clip Art pictures

- over Pictures per disk, 1000 images
- Picture viewer included with every disk
- Disk 1: Flowers
- Disk 2: Christmas pictures
- Disk 3: Space life & planets
- Disk 4: Animals
- Disk 5: Abstract pictures
- Disk 6: More general pictures
- Disk 7: Sports
- Disk 8: Buildings
- Disk 9: People & faces
- Disk 10: More animals
- Disk 11: Vegetables & Plants, green thumb
- Disk 12: Hospital C64 - more people
- Easily transferred to most disk drives
- ClipArtDisk pack disks £10.12 C64.95

Ideal for Label Wizard,  
PrintShop, PaperClip  
Publisher & Award Maker  
£23.95 each

### Disk Art

Exactly 1000 pictures for Commodore 64.  
These are not Clipart but large pictures,  
mainly in 2D.

- Disk 1: Holidays, general, weather, general
- Disk 2: Life scenes, like people, food, nature
- Disk 3: Industries, machines, P.C. screens, etc.
- Disk 4: Tools, gardens, animals, tools
- Disk 5: Maths & Science, computers, houses
- Disk 6: Animals, dogs, cats, space creatures
- Disk 7: People, baby stuff, health, health
- Disk 8: Office, household, computer fun
- Disk 9: U.S. States, countries, road vehicles
- Disk 10: Churches, cast & old house images
- Photo - Professional, ready to use forms
- MusicBox - Create music charts

High Quality picture disks  
£7.49 each

### C64 Graphic Software

Newspaper Clip Art disks £12.95. High quality pictures ready to use with Newspaper, full sentence per pic, each Pic £0.10.

Certificate Maker. Create certificates on the Commodore 64 and printout your Commodore printer £10.95.

Certificate Maker Library Disk 1. Add new borders & pictures to Certificate Maker. £10.95.

Business & House Maker. Make business, family packages. Home designs included £12.95.

Newsletter Creator. Super Newsletter disk allows creation, editing & printing news for your magazine. Ideal for Action Replay, Report, Power News, Disk 01.95.

### Graphic Label Wizard

Small size graphic label printing utility for the C64

- Print labels with font font & graphics
- Easy to use menus with easy to use controls
- Use PrintShop/Commodore Clipart pictures
- Offers graphics display - easy to print pictures
- Quality storage, layout & print your labels
- One graphic plus 10 lines of text per label
- Save labels to disk for later editing or printing
- Print multiple copies of your favorite label
- Print 1-100 labels at once
- 800x400 - 16 colour graphics

A flexible printing tool  
£19.95

### Newsroom 64

A desk top publisher for the 64

- Design, produce & print newsletters, posters
- Send faxes from disk output pictures
- Print press cuts, letters & other documents
- Photo Lab, system & disk output pictures
- Barcode, postscript message, banners
- Copy Board, word processor with 2 fonts
- Layout Board, quickly design your pages
- Edit package with up to 100 page master

A dynamic programme for  
journalists of all ages  
£24.95

### Award Maker Plus 64

If it's worth announcing, then it's worth  
announcing with an award they'll keep.

- Create beautiful awards with your 64 poster
- Ideal for any occasion, design your own
- Easy to use - just steps to create an award
- Feature up to many hours
- Award custom or photo-printing
- Choose a graphic from the large picture library
- Type in a message, name & add your signature
- Introduce your documents poster
- New feature - 8 point awards for whole classes
- Use the optional CDROM producer for borders
- Price: three reinforced prints on each

Create a keepsake they'll be  
proud to display  
£24.95

### Video Title Shop 64

Reshape your own Movie Director

- Use you C64 & add to old your home video
- Create an endless variety of titles effects
- Special effects (Zoom, pan, etc.)
- Paste one screen into another, move a screen
- Combine two & graphic effect & more

Publish your home movies  
£19.95

### Paper Clip Publisher

The most powerful Desktop Publisher for the Commodore 64

- Produce anything which uses words & pictures
- Newsletters, reports, presentations, thesis
- It's easy to use
- What you see is what you get shapes
- What doesn't measure to your results isn't
- Icons for commonly used tools
- Built-in font editor to prepare your text font
- Built-in graphics editor to create your artwork
- Millions of possible colours

- It's very powerful -
- Include multi-page documents
- Design layout for text or graphics
- Many layout options for the page
- Pages are automatically formatted
- Place text around artwork
- Import graphics from PostScript, Macintosh
- Use ClipArt pictures
- Import business word-processor
- Use columns, character spacing & margins
- Create image map graphics if fit pictures

- Create Graphics masters ...
- PhotoAlbum, photo album
- Open power edition, photo viewer
- Graphics reporter
- Textbox, bar graphs & fonts -
- 800x400, high resolution, dots, enhanced
- Shadowed, back, white, underline, justified
- Party - Individual, Times, Courier, Symbol

An impressive desktop  
programmer for home and  
professional use  
£34.95

### Solutions Unlimited

#### Photo Finish

- High/low Quality - switchable on your printer
- Options print titles, graphics, products ...
- Graphics with thousands the print resolution
- Compatibility with most imaging packages
- Print 1000's of photos in 10.0
- Photo Finish for the C64 C64.95

#### Image Factory

- Convert graphics from one format to another
- Interlace, PostScript, Photoshop, CompuPaint, Blazing Puddles, PhotoShop etc.
- Image editor - enlarge, shrink, invert, flip
- Crop and crop pictures
- Convert TIFF files for transmission & save
- Image Factory for the C64 £14.95

#### To Order

All products available by mail  
order, call now with credit  
card details or mail a cheque  
or Postal Order to the address  
below.

Prices include Postage & VAT.

Allow 14 days for delivery

# Oxford Basic

*Despite its name,  
this is a compiler with  
several very interesting*

By Norman Porte

The Oxford Basic package is a suite of four programs designed to get the best from Basic and beyond. The main part of the program is a standard compiler but, before using this section, the program has to be optimised.

At the beginning of the whole process, a good set of utilities can speed things up and keep it all neat. Oxford Base's *Toolbox* fits into the high memory slot at #40152, and offers ten new commands which are outlined below.

One glaring omission is an automatic keywarming command, and another is a command to undo the execution of a NEW or a REVERSE a program after a read. It's only my opinion, of course, but to me these missing commands would have enhanced its otherwise comprehensive package.

## Vital Statistics

The Analyzer really gets inside a program and figures out every possible value and measurement that the programmers could think about. It takes a little time for the program to be read in, but the wait is worthwhile, because there are seven options available for analysis afterwards.

A Flow Cross-Reference revealed the following facts about the program:

GENERAL INFORMATION		REGISTRATION	
GENERAL LICENSE		REGISTRATION	
1987	1987	1987	1987
1988	1988	1988	1988
1989	1989	1989	1989
1990	1990	1990	1990

The first number in the 400818 section is the line that is addressed and the five numbers that follow are the line which call or the recipient.

What is Data Cross-Reference is called, the variables are listed, followed

As a program is developed, it's easy to forget to delete lines which are no longer needed. The Deadwood Analyst cleverly scans through the program to find lines that will never be accessed. The offending lines are then listed, and a deadwood-free listing generated if required.

Another improvement can be made through the Create Header File option, which generates a short program to initialize all of the variables used in the program. The file is then saved and can be #include'd with the main body of the program later using the Toolkit. Initialization may seem as redundant as the LUT command, but it can actually improve the speed at which the program executes.

When a variable is encountered in a program, the computer checks through the list of existing variables to see if it's a new entry, or a modification of an existing one. If the label proves to be a new one, it is tagged onto the end of the current variable list. This involves a substantial degree of memory movement and consequently wastes time.

Added to this, a variable which is commonly used may be at the end of the list, which means that the routine has to check through the full list before the current value of the desired variable is discovered. The generated header program avoids these time wastes by reserving space for the variables in descending order of frequency right at the beginning of the source.

A list of program statistics can be displayed so that memory and variable usage can be recorded. The **Geometric Stats** option also summarizes other facts such as the number of lines used, QWORDS used, QWORDs as well as the total number of variables and arrays.

```

10 REM PRINT A SCREEN
20 FOR I=1 TO 100
30   PRINT I
40   IF I=50 THEN END
50 NEXT I
60 PRINT "END"

```

There is also a menu option for checking the disk directory, and to make sure that any files have been successfully saved.

The reports given by the status generator and the file and data cross-reference options can be used to check alongside the finished program. There is no direct hard copy option, but a utility file is included which will read a stored report to the printer for future reference.

### Arranged Crush

The final pre-compilation routine is the Compressor, which reduces the program to its absolute minimum size. As each line is scanned, the routine removes all REMs and spaces but the compression goes beyond this. Lines are combined to squeeze in as many commands onto a line as possible.

After compression, a portion of the program may contain lines which physically span three or more screen lines. This is possible because the Compressor works with one program code so that a command such as PRINT only occupies one byte rather than five. A line could therefore contain as many as 35 PRINT commands, which would expand to 225 characters and columns of five screen lines! This would make editing impossibly difficult, so the program must be thoroughly debugged before this operation is performed.

Although this optimises the program's memory usage, difficulties were experienced when using the compiler so I eventually decided that compilation was probably better used on programs which wouldn't be improved by compilation.

### Full Speed

OS-COMP is one of the best compilers that I've ever seen. It is a full-fidelity, fast pass compiler which can cope with all of the commands in the OS47 Basic library. The programs also

across all types of variables and arrays up to two dimensions. Three-dimensional arrays can be compiled, but only after the program has been pre-processed by a special utility which is included on the disk.

A compiler converts Basic code into a special form of machine code which runs up to 40 times faster than the original program. When a Basic program runs, the operating system has to scan each line in turn, take each command line as an action, and create and modify all of the variables it encounters.

A compiler scans through the Basic program and converts all of the commands and their parameters before the program is stored on disk. The conversion also means that sufficient space can be made for all of the variables which are to be used. The finished program is therefore midway between Basic and pure code.

The initial compilation can take a long time to produce, however the final program is scanned and re-scanned four times before the final program is produced. During these passes, the compiler produces and erases temporary work files which add disk access time onto the process. A long program can take up to half an hour to prepare, but the results are worth waiting for.

Worthy, the producers insist that the master disk should be used as little as possible, and a special back-up routine is included which renumbers the compiler and its associated utilities, cross-references data. Unfortunately, the back-up is not the most efficient of utilities, and each sub-program is transferred individually, regardless of whether it occupies one block or 20. The result is a constant swapping of disks, which makes the whole process very laborious and unnecessarily long.

Once the copies have been made, the compilation process is automatic, and the program to be processed is placed on the same disk as the copied compiler. This can cause problems with disk space, but this problem has been anticipated by offering variable size message generation and options to erase programs which will no longer be required for the final compilation — even the original program can be erased once compilation has progressed beyond the first pass.

The limitations of the compiler are surprisingly few and petty. Arrays must be definitely dimensioned, which means that DIM A(1) is a legal statement but DIM A() would be

illegal. If an array slips through to the compiler, which lacks an explicit dimension, the program generates compilation to ask for a suitable array size before continuing through the rest of the program. This may limit the efficiency of memory usage, but less programs would suffer appreciably by the insistence on explicit dimensioning.

The only commands that the compiler rejects are LIST, CONT, SAVE and RUN when it has a line number parameter. LIST is excluded because there is nothing to list in a compiled program. CONT is an acceptable omission and most people could live without commands such as RUN#0 since a CLR: RESTORE: GOTO#0 could be substituted instead. SAVE is harder to live without but this could be remedied by including an STS in the original program in a suitable machine code patch.

The only worrying problem which revealed itself under testing was that the maximum string length is 224 characters, instead of Basic's 255. This need easily be an oversight on the part of the programmers because it is not documented in the manual. Such errors are slightly disconcerting because they only show up at runtime, and I'd rather not do with my flagons crossed while a program compiles hoping that another unforeseen error occurs.

All compiled programs are executed with the RUN/STOP key disabled. In most cases this is unimportant but it may be necessary to allow such an interruption as a special form of REM statement has been devised to allow the key to be enabled and disabled.

### Error Tracing

If errors occur when the compiled program is being tested, it can be difficult to relate the problem back to the original program. An error which stops the execution of the program will generate a message which is similar to Basic's equivalent message, but instead of a line number a memory location is displayed.

A utility on the disk can be given this value, and it will then automatically cross-refer this and indicate the corresponding line number in the original program to assist debugging. Of course, such an error should rarely occur because any sensible programmer would have tested and

debugged the original program before  
compilation, would it then?

Another utility which can aid debugging was actually devised for hybrid programs which require parameters to be accessed by a machine code patch. Variables and arrays are stored differently in the compiled program so any associated code routines have to be modified. This works in favour of routines which access a current variable value, because the memory location of variable data is fixed in a compiled program. REPORT is a utility which scans the original program and prints out a table of variable locations to ease the problem of conversion.

## **Party Protection**

Why, in this day and age of colour photocopiers and cameras, do publishers insist on these ridiculous colour charts? I don't mind protection systems, but I object to this form because I am red/green colour blind and the responses always seem to insist on using red, green and blue often

colours on these charts. Why black, white, red and yellow can't be chosen defeats me.

In this case the problem is increased because each of the four programs is individually presented. In the end, I got so annoyed that I resorted to backing up the programs using a cartridge copier - so much for moderation!

Despite this, the package maintains a very high standard throughout all four sections. Maybe if the space under the ROBOTS had been used, the toolkit could have been more comprehensive in the manual, but no compiler is perfect and this compiler is comparable, though slightly inferior, to Superopt's likes. What this has to offer which Brix doesn't is the other three packages, which makes this package stand out within the competition.

### Toolkit Commands

CHANGE  
DELETE  
DUPLICATE  
END

INFO  
ABOUT  
QUIT  
RENUMBER  
SEND  
TYPE

#### Analyze Options

[File](#) [Print](#) [Close](#)  
[Edit](#) [Cross-references](#)  
[Format](#) [Disk directory](#)  
[Deadwood analysis](#)  
[Create header file](#)  
[Generate stats](#)  
[Report to Excel](#)

### *Comments on Functions*

Removes spaces and HTML tags  
*(Comments are removed later)*

### **On-Course Evolution**

Majority of Basic 2.0 commands  
Four pass operation  
Error checking  
Location of variables printed

AT LAST A POOL'S PROGRAM THAT BELIEVES THE GOOD!!

*PROBLEMS OF THE STATE*

THE PROGRAM THAT GAVE HUNDREDS OF DIVIDENDS TO ATARI ST USERS, IS NOW AVAILABLE FOR THE COMMODORE 64. POOLSHUSTER IS QUITE SIMPLY THE MOST ADVANCED POOLS PROGRAM AVAILABLE TODAY. LOOK AT THESE HIGH-SCORING FEATURES: (1) POOLSHUSTER IS GUARANTEED, THAT'S RIGHT - WE'RE SO CONFIDENT THAT YOU'LL WIN WITH POOLSHUSTER THAT WE PROMISE TO REFUND THE PURCHASE PRICE IF YOU HAVEN'T WON SOMETHING WITHIN ONE YEAR OF THE DATE OF PURCHASE! (2) THE STATE-OF-THE-ART POOLS PROGRAM, IT USES AN ARTIFICIAL INTELLIGENCE (A.I.) SYSTEM TO FINE TUNE ITS PREDICTIONS EACH TIME YOU ENTER A SET OF SOCCER RESULTS. IT ACTUALLY LEARNS FROM THE RESULTS IT GETS WRONG. (3) & IT KNOWS THE SCORE! POOLSHUSTER CONTAINS A MASSIVE DATABASE OF SOCCER STATISTICS WITH DETAILS OF OVER 1000 PAST MATCHES. & IT'S EASY TO USE. POOLSHUSTER IS FULLY JUSTICE HOUSE DRIVEN - THERE'S NO NEED TO USE THE KEYBOARD AT ALL. SO IT'S VERSATILE. POOLSHUSTER COMES WITH ALL THE UK SOCCER LEAGUES YOU'RE LIKELY TO NEED, INCLUDING GM VAUXHALL, NORTHERN PREMIER, BEASER & HIS LEADS, AND YOU CAN ADD ANY OVERSEAS LEAGUES AS YOU WISH. AT YOUR FINGER TIP, THE STARS POOLSHUSTER ALSO INCLUDES THE UNIQUE MAJOR PREDICTION PROGRAM. THIS FORECASTS LINES OF 16 POSSIBLE DRAWS ACCORDING TO THE ASTROLOGICAL POWER NUMBERS FOR YOUR NAME, DATE OF BIRTH & POOLS DATE. POOLSHUSTER IS AVAILABLE ON DISK OR CASSETTE.

**POOL BUSINESS COSTS ONLY \$40.00 SO HURRY  
ORDER ONE TODAY & WIN THE POOLS TOMORROW!**

```

10 PRINT "PROG"
20 PRINT "RUN IT"

40 GOTO 80

50 RUN
60 LINE MISSING! HELP!!

```



# Line Input

*Improve your 64's input  
with this handy utility.*

**T**here are times when the standard Basic keyword "INPUT" on the C64 is a bit of a pain. In the event that a comma (",") needs to be input as part of a string "INPUT"

cannot do it, because the comma is interpreted as being a delimiter between different lots of data which are supposed to be assigned to separate variables.

If you wanted to enter such a string then it has to be enclosed by quotes so that "INPUT" will read in every character. This looks silly on the screen, and is very confusing for users

of the program who are not familiar with the technique - they can't understand why commas has to be enclosed in quotes in order to enter strings including commas, whereas other forms of input such as numeric data do not. It's even more confusing when the program later requires that the user use the commas to separate input into different fields!

The other slightly aggravating thing about "INPUT" is the fact that the ":" prompt is always printed on the screen prior to "INPUT", waiting for keyboard entry. This is sometimes a nuisance, since the message which asks for the input on the part of the user is not always of a kind where a question mark at the end is relevant - that is, it's not a question. Add to this the need to follow it with quotes - and then replace one and type quotes again so you can use the cursor move keys to edit the line - just so a comma can be included where required in the input string, and straight away we already have two irrelevant characters on the screen before anything is actually typed in.

What's more, you always forget to use the opening quote at the start, which just suggests that it doesn't feel natural to have to do this. The situation is worse when inputting from an external channel - here not only the commas but also the colons and semi-colons are interpreted as carriage returns, or the end of the string, whereas in actual fact it may be nothing of the sort. Again, enclosing the string in quotes before it is saved to the external device prevents this, but on inputting it back the quotes will have disappeared. In either case, whether inputting from the keyboard or an external device, quotes cannot be included as part of the text. But why not, you may ask?

"GET" will input anything from the keyboard, and a string comprising all printable characters can be built up using "GET". But the Basic coding required to continuously add the string at the same time rapidly becomes wordy and slow. Also, the input is not automatically echoed to the screen, so this must be done in Basic. There's no flushing cursor either.

A cursor can be forced, but it has the peculiar behaviour of leaving some characters in inverse video as soon as the cursor move keys are used. "GET" can be used to input text from an external device, but using Basic commands that retrieve one character at a time and build it into a string can become-consuming to say the least. It's frustrating when all you want to input some text of an unusual format.

If you're a machine code programmer of any kind, then this can be got around, except that you may find yourself having to re-develop the coding to suit the requirements of different programs. But if you want

to stay in the Basic environment, then while "GET" and "Input" have their place (and are very good at what they do), what is really needed is a new Basic keyword, one that will input strings of all printable characters.

Well here it is. Inspired, as it happens, by the IBM PC BASIC A's action keyword of the same name the utility is called "LINE INPUT". It is in the form of a machine code routine stored at \$C0A0 to \$C0B0 high up in the free RAM area above the 64's BASIC interpreter. In actual fact, it is not a real keyword at all, but a "SYS" call which claims to be pretending to be an interpreter routine.

This is necessary because of the loss of information that the routine needs in order to carry out its task. The "USR" function only allows one numeric value to be passed on to a machine code routine, which then can only return one other value to a variable. The temporary register stashes for "SYS" calls are rather long-winded to use, and in any case none of these are any use because "LINE INPUT" must return a string as a string variable.

The "simple answer" is for the routine, "LINE", to get these parameters direct from the current Basic line being processed using the subroutine "CHRGET", which retrieves the next BASIC byte into the accumulator, and also incidentally is the thing that skips spaces and/or quotes - in exactly the same way as all the interpreter routines get their information. This makes "LINE INPUT" very easy to program in Basic, because if you assume the presence of the word "SYS" at the front it will appear in the listing, and in operation, like a valid Basic keyword. Furthermore, if you make a numeric variable equal to the "SYS" call address and name the variable "LINE", the subroutine will be compatible. The syntax then is as follows:

```
STS < address > INPUT
I < channel > N> < string
variable >
```

Note the full stop between < address > and INPUT(1). This is very important as it ensures that the following valid BASIC keyword, "INPUT" or "INPUT A", is crunched down to its proper BASIC state when the BASIC line which contains it is entered. Neither of these will be executed by the interpreter in the normal way. However, they exist here exclusively

for use by "LINE". In fact, the processing activity of "LINE" will make whichever of these two tokens inaccessible to the interpreter.

The keyword for "INPUT" or "INPUT A" is included because "LINE" has to vector within itself to the appropriate processor for either inputting from the keyboard and screen, or from an external device. The keyword is tokenised for three reasons 1. to save memory space, although this will be negligible, 2. because it's a professional way of doing it, 3. since "LINE" won't be told which form of input is required, it simplifies things greatly if only one byte can be assumed instead of a string of characters. This token, and the remaining parameters, are read by "LINE" direct from the Basic text using "CHRGET".

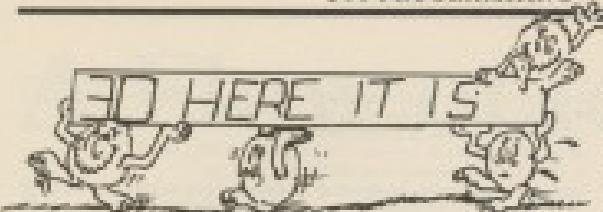
A number of subroutines are used by "LINE" to read the parameters. They themselves get the basic bytes using "CHRGET". One of these is a cascaded subroutine called "GETVAL" located at \$C0B7 (\$2135 decimal), and looks like this:

READ	EQU	SADRA
FIX	EQU	SEPF7
ORG	EQU	SEB#7
GETVAL	JSR	READ
HPIP		FIX

"READ" is part of a Basic interpreter routine, which picks up a numeric value from the Basic line currently being processed. The value may be written as a string of decimal numbers, or be in the form of a named variable. Whatever, "READ" converts it into floating point format and stores it in Floating Point Accumulator #1.

"FIX" is the familiar "fix to fix" interpreter routine. The constants of PAC... are converted into a 16-bit integer and stored in page 0 area \$14-\$15 hex.

Therefore "GETVAL" performs the action of reading a numeric value as a variable's contents from a Basic line and making same available in handy double-byte integer format in \$14-\$15 for a machine code to use. If "GETVAL" is handled properly, it helps in understanding how "LINE" operates, that the C64's Basic interpreter follows a special convention that every interpreted routine, including those used by "GETVAL", expects the first Basic byte that it has to deal with to already be in the CPU's accumulator. Or, to put it another



way, every Basic interpreter routine calls "CHRGET" to get the next byte of Basic text into the accumulator before it exists. Also "GETVAL" is a little bit limited, only positive numbers in the range 0-65535 are allowable otherwise an "ILLEGAL QUANTITY" error is generated. Invariably scanned numbers produce "SYNTAX ERROR".

On calling "LINE" with the "SYS" command, true to the convention, the interpreter runs "CHRGET" to load the accumulator with the next byte of Basic text. Unfortunately this doesn't arrive intact when "LINE" takes control, because the action of successive "SYS" loads the accumulator with the contents of the temporary storage for A reg.' location at \$030C (00 decimal). So the accumulator has to be refreshed with a call to "CHRGET", a letter part of "CHRGET" which gets the same BASIC type again. This byte must be a full stop (".") or \$2E if it is not a jump to L1090 which prints "SYNTAX ERROR IN LINE<n>". and terminates the Basic program. The byte following this, retrieved by "CHRGET", must either be a token for "INPUT" or "INPUT#". If it is neither of these again a jump is made to "SYNTAX ERROR".

Otherwise at this point the routine "LINE" deviates. The bulk of the routine comprises two separate processes, the one chosen depends on whether the input is from the keyboard and screen, or whether it is from an external device.

#### LINE INPUT from keyboard and screen

**Syntax:**  
\$118 < address >; INPUT < string variable >

#### Example:

```
100 LINE : $118
100 SYS LINE,INPUT,AS
```

Input from keyboard and screen is carried out in three stages. Firstly the screen RAM address for the start of

the input is found by locating the cursor position with the kernel routine "PLOT", and calculating the screen address by the rather crude method of adding 40 to the screen base of 1024 "y" (row count) number of bytes. "z" (column count) is added to the result. I did originally write a proper initialization routine, but this required so much code that it wasn't really worth it. The calculated address is stored in free-zero page locations as a pointer.

All you are on the screen is a flashing cursor with an "FF" prompt. Now a kernel routine called "CHRINP" handles all the character getting and printing with full editing support and proper flashing cursor. In use it is identical to using the Basic editor in insert mode. It has the same limitations - if quotes are typed then cursor move keys come onto the screen as characters, but ridding em quotes and rekeying stops this. "LINE" measures the length of the line. This is done by searching backwards from the maximum string length position, the details of which is \$0, looking for a non-space character. On finding a non-space character, its position ahead of the start of input becomes the new string length.

The screen line, from the start address up to the string length, is copied to a buffer. This is our old friend, the discrete header buffer at \$030C. However \$030C is reserved to hold the string length count, so the buffer actually starts at \$0305. A conversion routine is used to translate screen coded characters into CBM ASCII, and the most significant bit is stripped off in case the screen characters were inverse video ("LINE INPUT") doesn't input a line which has just been typed, it could just as easily have been PRINTED instead and the keyboard buffer POKE'd with a carriage return).

At the end of this the string is stored in the buffer and its length in \$030C. At the final stage "LINE" calls interpreter routine "LOCATE", which goes in search of the string variable through the variable storage

area - "LOCATE" gets the variable's name from the Basic line. If it does not exist then it is created. After this the string storage for the variable, if already in existence, is freed (by "FREESTR") and then a fresh storage area is defined using "STRRES" which does so according to the new string length passed to it. We now have a place in the string storage area to which the contents of the buffer can be copied.

It is impossible for characters to be put in this area directly (without being stored in a buffer first) because "LINE" does not yet know how long the string will be until this stage.

In this manner any string variable can be created or updated using "LINE,INPUT" in just the same way as it can by any other INPUT function. However some care is required where the variable is one of a dimensional array. Small arrays up to a "safe" limit of say 10 or 11 elements would be okay, in other words

#### 110 SYS LINE,INPUT,TW0

Where "W" is an index, from 0 to 10 in the default provision for arrays up to 11 elements, but otherwise a "working variable" such as INS for example would have to be used to transfer the input to the element where this is one of a large, dimensional array, e.g.:

```
110 SYS LINE,INPUT,INS
120 A10w = INS
```

If this precaution is not observed the process becomes too complex for the simpler than normal string handling techniques used and usually causes the sit.

Whatever the string is stored in the proper place, and the extra does not add perceptively to the processing time, even if a large string area is being filled with text.

Because "LINE" is located in RAM it can be modified. I mentioned that for input from keyboard and screen, it has a default maximum string length of 80 screen characters (two screen lines). It doesn't matter in which column across the screen the start of input occurs, the maximum remains 80 characters. You can change the maximum length by a POKE to suit your own requirements. For example, a disk file name cannot exceed 16 characters in length. If "LINE,INPUT" is used to get the file name it can then be modified only to accept

up to a maximum of 16 characters. In the source listing the maximum string length is labelled "MAXLEN" and it is \$C0A1D (\$3045). Use the following:

```
100 LINE = 51004
1000 MAXLEN = 120#H
1010 POKE MAXLEN, 16
1020 PRINT "ENTER A NAME
FOR THIS FILE"
1030 SYS LINE INPUT INS
1040 NAME = INS;"$M$"
1050 OPEN 2,2,2, NAME
1060 END
```

You can type as many characters as you like for the file name, only the first 16 will be taken, no notice of after you press [RETURN].

Be aware though of a couple of limitations of "LINE.INPUT". One is the same as that of the normal "INPUT", namely if other characters present on the screen come within the range of the maximum string length, it will be assumed that these constitute the end of the string, even if you don't want them. Though blank space does exist beyond the position where input commences to prevent this.

Also, "LINE.INPUT" doesn't know if the start of input occurs on the bottom line of the screen. If the screen scrolls up where text exceeds one screen line, the start address is not adjusted up with it. Also "LINE.INPUT" will assume that the rubbish beyond the top of the screen RAM area are characters to be input! Consequently "LINE.INPUT" should not be used any lower than the 24th row down.

LINE INPUT # from an external device -

#### Syntax:

SYS <address>, INPUT < character No. >, < string variable >

#### Example:

```
100 LINE = 51004
```

```
1000 OPEN2, 2, 2, "TEXT"
1010 SYS LINE INPUT 2,AB
1020 END
```

This is the only time that "GETVAL" is used, and even then only the least significant byte of the resultant integer is needed.

"GETVAL" retrieves the channel number following "INPUT # so that

internal routines can read input from the appropriate channel. Inputted characters are copied to the buffer, a process that continues in a loop until one of three conditions are true:

1. The ST variable indicates an EOF signal from a disk drive. On detecting "EOF" inputting is terminated.
2. The character read in is found to be the termination character. The default termination character is a carriage return (\$13), an encountering the termination character inputting is terminated. The termination character is NOT copied to the buffer.
3. The maximum number of bytes have been copied to the buffer. The default maximum number of bytes is 128. On the 128th character being copied to the buffer, inputting is terminated.

On input being terminated, "LINE.INPUT # " goes to the final stage of building the string to memory exactly as before. As far as "EOF" is concerned, it will be required that the BASIC program using "LINE.INPUT # " should also monitor the "ST" variable. Reusing "LINE.INPUT # " and forcing input past EOF causes a rubbish character to be copied to the buffer, overwriting the previous string. This is not a problem if other variables have been made equal to this string.

As before, because the routine exists in RAM, it can be monitored to behave in a special way. "CHCR" is a label in the source listing which represents the point where the character read is tested to see whether it's the termination character or not. It looks like:

#### CHCR CMP # \$0D

Where the operand is the value, in this case 13. This can be changed as in the following example:

```
CHCR = $2003
POKE CHCR + 1,B
```

Now the routine will stop reading when it encounters a zero byte. Similarly the maximum number of bytes copied to the buffer can be changed:

```
CHRMAX = $2009
POKE CHRMAX + 1,<n>
```

Where <n> is any number up to a limit of 191. Because this is the maximum amount of space available in the cassette buffer area.

Even better, "LINE.INPUT # "

can be made to ignore any termination character. Following the "CHCR" location is this:

#### ISTERMREQ.PUTLEN

which is where a branch is made to the final stage upon encountering the termination character. By using

```
ISTERM = $2003
POKE ITERM, 234 : POKE
ISTERM = 2,234
```

This has the effect of overwriting the branch instruction with "NOP\$T", so the routine carried out if a termination character is found. This allows maximum flexibility for inputting data of a "strange" nature - "LINE INPUT" can read all byte values 0-255. It is possible for example to read machine code into string - suppose you wanted to do such a thing!

Because "LINE.INPUT # " is completely self contained, i.e. no part of the conventional interpreter INPUT # routines are employed, we are allowed to do something normally quite illegal. The following is possible in fixed mode:

```
LINE = 50994
```

```
READY.
OPEN 2,2,2, "TEXT"
```

```
READY.
SYS LINE INPUT 2,AB
```

```
READY.
1AB
```

This is incredibly useful for verifying that a Basic routine that you are trying to de-bug has saved data properly or not. It's normally impossible to get at this data in direct mode, since using "INPUT # " results in an illegal direct command error. You can go on moving the cursor up and down printing and displaying successive fields of data. This even works with numeric data saved using "PRINT # ", since these are written as decimal strings.

If you have an assembler you can enter the source listing, and if needs must the ORG statement can be changed to relocate the routine anywhere you like, but make a note of the new label location. If you then want to customise it with POKEs, it's unlikely that you will need to use "LINE.INPUT # " very often, but when you do, you'll be glad it's there.

## A Flow of Ideas

It's often necessary to view a directory from within a program, but from Basic this can be very difficult. There is a way, and this method can reveal more information than may at first be obvious.

Before a disk directory program can be written, it is essential to know what data the drive makes available. Using the following short program on the directory in Table 1 gives all of the information which is displayed in Table 2.

19. 1990-1991  
20. 1991-1992  
21. 1992-1993  
22. 1993-1994

The recorded data includes the Block Allocation Map (BAM) which key from \$0000-FFXX, as well as the directory content itself (0XXXX-XXXX). Using GETw, each byte can be read and used to form quite a powerful source of disk information.

From this readout, the name of the program can be derived. What we will create is a flow diagram for a program which lists each directory entry, the file type, blocks used and the track and sector (the specific block location) relating to each file. As the program runs, it will keep track of the total number of programs on the disk for a general round-up screen, which is to be displayed after the individual program detail screens.

The program can also keep a tally of the total number of blocks used, and this value can then be compared with the number derived from the RAM to ensure that the disk has been validated correctly. Finally, a detailed map of the number of free sectors (blocks) on each track can be displayed using colour to differentiate between tracks which have not been used at all, and those which have lost a few sectors to file storage.

The first duty of the program is to set the screen columns and then initialize the disk under examination. This means opening command channel 13, and keeping it open for checks on disk errors throughout the program.

Next, the directory file is opened for a sequential read operation. After the driver loads the first sector into its

### *How easy is it to incorporate a disk directory reader in a program?*

#### **The Vacuum Pump**



internal buffer, variables are initialised and dimensioned ready for the main program to begin. The reading of the RAM takes a little time, so a comforting message is displayed to assure the user that this is a normal operation.

### **Excluded BAM**

The layout of the BAMI can be seen from Table 1. There is no single byte pair responsible for recording the number of free bytes on the disk, so this value has to be derived from the data at the beginning of the BAMI. The first two bytes (the track and sector links) are not loaded when the directory is read as a sequential file, and the next two bytes are irrelevant, but some of the next group of bytes are essential to calculating the number of free memory blocks on the disk.

The information is stored in a specific map, with the bytes grouped into clusters of four. The first byte is the total number of free blocks on a particular track, and the three bytes that follow can be used to calculate which particular sectors these are. In this case, the only bytes of interest is the total number of free blocks, so that is read and added to a running total while the others are discarded.

Apart from keeping a record of the total number of 100 blocks, such

track's details must be stored in an array if the aim of producing a track map is to be fulfilled. There are 10 tracks on a normal Commodore disk, and it can be seen from Table 2 that there are, in fact, 16 groups of four bytes in the RIM. A loop must read each of the groups, storing the first byte in an array and adding its value to a grand total of free blocks. After this is done, the next three bytes can be ignored.

At the end of the BMT, the disk name and ID numbers can be found. This must be read and stored in a suitable form for display at the head of each screen page. First of all, the string is converted to produce a label indicating that what follows is the disk's title and, to comply with Commission's convention, the BPS/ON character is added to display the disk header in extended characters. A second convention is to place the disk name in quotes. This is not carried for those reading the directory file, as the opening quote is added at this point.

The disk name is allocated a space of 16 characters in the RAM sector of the directory. This means that by simply reading the 16 character group and adding it to the file string, any disk name can be caused for without any complicated checks. The resultant string is then completed with a closing quote mark.

To complete the screen page header, the next 13 bytes are added to the string. This actually reads more than enough characters to cover the ID and the disk type descriptor (always \$B0) on a normally formatted disk, but is extended to cover for non-standard formatting using four character IDs and other such tricks. The difficulty here is that some of the bytes will be stored as 'null' values', which is the reason why using variables are being used in the program instead of hardcoded ones.

Because the bytes are stored as ASCII values, the conversion routine must look for these nulls and convert them to CHR(0) to avoid similar errors when the bytes are converted for use. This happens quite a bit, and a subtraction is the ideal answer. Once the disk name has been converted as

a string, it is printed and, because this string contains the clear screen symbol, the routine acts as a title for the screen page.

Before reading in the directory entries, the system tracks must be displayed. The convention that has been devised for this routine is that the program name will be followed by the file type, block count and the track and sector values for the first program block.

This information is not stored in the correct order as it must be read, stored and sorted out in a print statement. The layout of a single directory entry can be seen in Table 4. A subroutine deals with the actual reading of the individual entries, and this will be examined later.

As each directory entry is dealt with, a counter is incremented so that a check can be made for a full screen page (20 entries). When this number is reached, a 'Press Any Key' message is printed and a keypress detection loop initiated.

Directories come in all shapes and sizes, so the program cannot work on a simple loop to read in the entries. A way is needed to indicate when the directory is complete. This is done by reading the system variable, ST. When this has a value of zero, work is still in progress, but if it has a real value one of two things has occurred.

It may just be that the directory has been read in completely, or it could be that an error has occurred. Before closing the file, the routine checks the error channel. If an error is detected, the program will fail and the message is printed; if there is no error the program continues to the next stage.

Under a screen page title of 'Visual Statistics', the details of the number of programs, and blocks used are printed, suitably labelled. Both are derived from the individual directory entries and supplied via the 'read entry' subroutine.

The 'free blocks' value has been calculated from the RAM record, but before this is printed we can use this value and the 'blocks used' value to check for a correctly validated disk. If the number of blocks used is subtracted from the total number of blocks free on a newly formatted disk (198), the result should equal the RAM-derived free block count.

Any inequality means that the RAM is faulty, and this is flagged by a suitable message. A word of warning - inequalities could be the result of

REl, or USR files being used on the disk, so ensure that this is not the case before validating, or you could lose valuable data!

After the free blocks have been displayed, a table is drawn up with the tracks listed above their usual sector values. As each entry, these values are checked against the sector capacities for each track. If none of the sectors have been used, the value is printed in bold, but any track which contains file data are highlighted by using light blue characters.

This program can easily be extended and improved to give a full RAM map, or reduced to produce a disk-to-screen directory printer.

The subroutine to search each directory entry is an integral to the running of the main program and, therefore, worth a closer look.

## Directory Delivery

The directory read routine takes over immediately after the disk name has been read from the disk. The directory details are preceded by a series of null bytes which must be discounted until a 'real' byte value is read in. This method saves time but has one serious drawback when the first file has been deleted and not replaced.

File types are denoted by byte values as follows:

```
129 = STD
130 = PRG
131 = USR
132 = REL
```

If the file is 'locked', or protected, these values are increased by 64 to give a range of 193-196. Deleting a file simply results in these values being replaced by a null byte. Using the method of discounting null bytes would mean that the first significant value would be the old track number when a deleted file was encountered.

Consequently, the program includes a check to see if the value read in is excess of 63. If this is not true then it must be a deleted file, so 'REL' is assigned as the program type and the routine jumps to evaluating the read in value as a track number. If the value is valid for a file type, a check is made to test if it is locked or not. Provision is made to denote a locked file by using a reversed 'less than' symbol, which is stored as a string. If the file is unprotected, this string remains as a space.

A second string is derived from the file descriptor value. This is the file type which is calculated directly from the file type value in conjunction with a MDIR statement. After the new type has been read in, the pathway which the DDIR file option took converges with the main program so that the track value can be assigned to a string before the sector value is read and similarly stored.

The disk name had 16 bytes assigned to it, and the same is true for the program name which follows the sector byte. A loop reads this in and concatenates a string. If a program name has less than 16 characters, the entry is padded out with shifted space values (190), and this is useful for formating the screen printout later, so they are not discarded here.

After the name there are nine byte types which are read in and disregarded, so the next significant bytes denote the number of blocks that the program occupies. This value is stored in low byte/high byte format and a suitable routine is included to reveal the true decimal value. All that remains is to print the information out in the correct screen columns. Deleted file names are shown in orange and existing files in white.

One problem with filenames is that they sometimes contain control characters such as screen clear, case switch or even colours. To avoid these marring up the beautifully formatted display, location 212 is polled with a value to fool the computer into thinking that it is in quoted mode, so that the reversed character is printed instead of the action which it represents being executed. Once the run of the information is printed, control is then returned to the main program.

## Directory Flow

The flowcharts show the logic of the program contained in the Listings pages. This is rendered as a Basic routine, but the beauty of flowcharting is that the same logic pathways can be applied to machine code or just about any other language that has been invented.

If you decide to investigate directory reading further, you'll also soon realize how useful the charts are for modifying the program. It's far better than wading through reams of listings, noting variables and then trying to make sense of it all.

THE ULTIMATE BACKUP CARTRIDGE GOES ONE BETTER...

# NOW ACTION REPLAY Mk V PROFESSIONAL

ONLY  
£34.99  
POST FREE

AR-V  
TAPES OR DISKS  
20MB TO 100MB  
8 Seconds/Sec

FOR CBM64/128 HAS ARRIVED



EVEN MORE POWERFUL, MORE FRIENDLY & NOW EVEN MORE FEATURES!

ALL FEATURES AVAILABLE TO TAPE OR DISK  
USERS.

ACTION REPLAY Mk V features over 20 more cartridge features in addition to the 100+ found with its predecessor including:

8 Seconds/Sec

• ALEXANDER DESIGN: includes an additional technology which greatly speeds up seek times and reduces seek times by up to 50%.

• DIRECT ACCESS: allows you to move to any location plus you access material

DIRECT ACCESS, DIRECT LOAD/SAVE, then copy other cartridges. There really is no comparison, there's just one of our machines...

• TAPING: now includes a feature that allows you to move to another cartridge and tape it directly onto the new cartridge. • DIRECT LOAD/SAVE: allows you to move to another cartridge with ease - especially if you're using the new cartridge.

• DIRECT ACCESS: allows you to move to any location plus you access material

• DIRECT ACCESS, DIRECT LOAD/SAVE, then copy other cartridges. • DIRECT ACCESS: allows you to move to another cartridge and tape it directly onto the new cartridge. • DIRECT LOAD/SAVE: allows you to move to another cartridge with ease - especially if you're using the new cartridge.

• DIRECT ACCESS: allows you to move to another cartridge and tape it directly onto the new cartridge with full range of commands.

## PERFORMANCE PROMISE

Action Replay has an unmatched range of standard features. When you buy Action Replay, if you don't find one feature or two, there are over 100 days free to tell us about it.

## GRAPHICS SUPPORT UTILITIES DISK

DISK ONLY. Price some graphics programs to add to your system.

SHOW OFF! It's easier to show you can make any part of your pictures in full colour, than it is to tell anyone else.

GRAPHICS EDITOR: A complete graphics editor. Edit your own graphics or edit graphics.

MESSAGE MAKER: Create your own messages, send them to anyone or edit them.

ONLY £12.99

## NOTICE 1988 COPYRIGHT ACT

© 1988 DATTEL ELECTRONICS LTD. ALL RIGHTS RESERVED. NO PART OF THIS CATALOGUE OR ITS PRODUCTS FOR REPRODUCTION OR DISPLAY MATERIAL, IN WHOLE OR IN PART, CAN BE USED WITHOUT THE EXPRESS WRITTEN CONSENT OF THE COPYRIGHT OWNER OR DATTEL ELECTRONICS LTD.

## TOTAL BACKUP POWER

The most powerful built-in cartridge ever devised. Unparalleled backup speed. Faster at any point. And prices like major systems in building these programs.

• 10MB TO TAPE • 10MB TO DISK •

• 10MB TO TAPE • 10MB TO DISK •

10MB has a dedicated cartridge designed specifically for cartridge tape. Backups (estimated 100MB/100MB) of cartridge, for "program" tape or tape formats required.

more processes to synchronize - 10MB TO TAPE AND DISK • 10MB

## SMALL LINE BUFFERED TAPE

"The smallest, fastest and least expensive". This is truly the best value for money cartridge. • The Cambridge Range™ • Cambridge Disk Drive

## UPGRADE INFORMATION

10MB Professional to 10MB Professional - same unit can be used as an upgrade from a new 10MB Professional version. Disk has price for just plugging into an enhancement card to "enhancement" - small price and cartridge price adds to cost, and upgrade of Cambridge Disk Drive.

ALL ORDERS POSTALLY & TELEPHONE WITHIN 48 HRS

## HOW TO ORDER ...

### BY PHONE



0182 744700

100% Credit  
Guaranteed

### BY POST



Send Payment, Full Name  
and Address to:

DATTEL ELECTRONICS LTD.  
100% GUARANTEED  
100% CREDIT GUARANTEED

### FAX

0182 744200

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE OR RECALL AT TIME OF PURCHASE  
AND SUBJECT TO CHANGE WITHOUT NOTICE.

EXCLUSIONS EXCLUDED: Please enclose quote by telephone prior to mail.

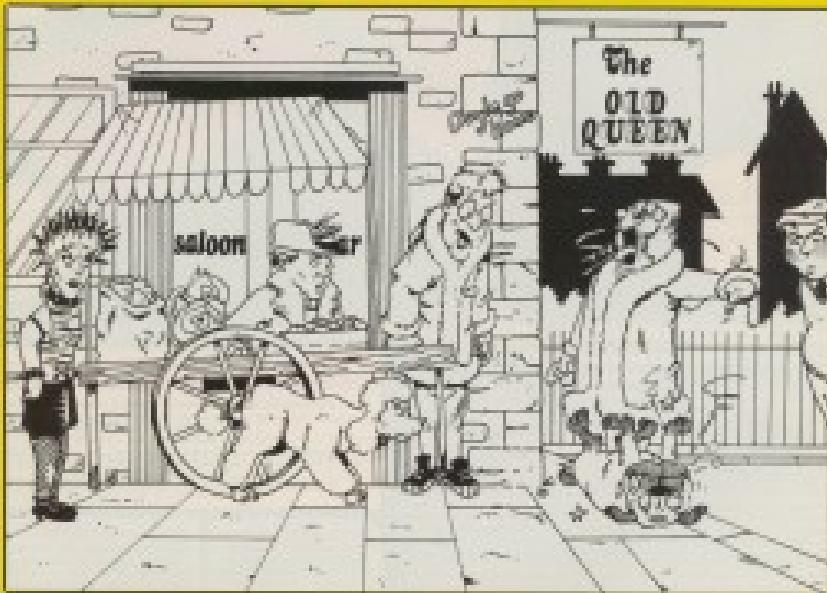
**DATTEL**  
**ELECTRONICS**

DATTEL ELECTRONICS LTD., FINTON INDUSTRIAL ESTATE,  
COVENTRY ROAD, FINTON, STOCKPORT, ENGLAND.

TELETON 0182 744700

TELEFAX 0182 744200

# • Deadenders



# War in Middle Earth

The world of Tolkein's Middle Earth once again comes to the computer screen only this time as a war game. In it, you control the forces of the Fellowship of the Ring in a desperate fight against the evil forces of Sauron. Your objective isn't military victory, it's far more important than that, as you must clear a path through the evil forces for a hobbit to take a Ring to the cracks of Doom.

At your disposal you have the dwarves of the Iron Hills, primarily elves and men of good heart as well as the Fellowship party of Frodo, Gandalf and Aragorn. Frodo is, of course, the ringbearer and his wear is at will to escape the battlefield and the ones. However, it has its dangers as wearing it corrupts the soul and takes you into the realm of the Nazgûl who will attack you if they are near.

The main display shows the map of Middle Earth and a finger pointer with which you can move in any area and view the forces in it and issue orders for them to move into battle. Eventually, forces will clash with the undisplayed

# GAMES UPDATE

**D**ouleover is an adventure based on everyone's favorite soap opera - Neighbours. Not seriously gay, it all takes place in Herbert Square, home to the Sopras. Foulings and other unscrupulous characters.

The story starts after 1001 has left the Wille. Yes, in case you haven't guessed, it's a soap full of caricature and one or two slightly smutty jokes.

You play the part of PC Donald Dancer and your mission, should you choose to accept it, is to find out who did the dirty on Willie. The baby tells you that poor old Willie turns up squashed, and with two holes in his neck. Even if you think the Sopras will never believe this, it also goes on to say that you can talk to the characters that you meet. "Ask Bandy about Color" is cited as an example. I tried it, and was met with a "You speak, but it doesn't change your inquiries." None of the Deardeners seemed to talk to me. Was it something I said?

The jokes are obviously aimed towards Australian fans, but it does give believe that non-dwellers of the series would get at least a smile from the game. Most of the humour, for me at least, came whenever I examined a character. When I examined Stevy, I was told "Purple eyes and bright green hair, the resembles a technician's mask." Forstall. All the descriptions are along those lines, and most are just plain daft.

When it comes to gameplay, I have a few reservations. The game doesn't accept some of the normal abbreviations. I had to type in "bathroom" as opposed to just "B". Not all the exits were listed, even the ones that should have been obvious, and mapping left something to be desired. When standing at the landing of the Old Queen, I was faced with the kitchen, bedroom and living room, yet no directions were given. It was a case of "Enter kitchen" etc., and no matter which of them I went into, the exit was always to the stairs. This system wasn't used consistently, though. From Herbert Square I could see the pub, the cafe and the corner shop. This time I had to input the word

N, S, E and W, and hope I went to the building I wanted. Some things should be granted in an adventure and sticking to one way of signifying directions is an agony.

During the course of the game, I came across two bugs, one funny and one annoying. I'd got a kiss from Ante and, whenever I tried to examine it, the game crashed. However, I was allowed to "search her" with no ill effects. Later, I discovered that I could "wash her" but only if I wasn't carrying it. I can just imagine how frustrating it would be for someone to find that out after going for a long time with no saves! The other bug was a silly one. I'd been walking through a station, as one does, and my investigations took me to the latrine where I was reminded how disgusting my uniform was. No problem, I took it off, washed it in a machine, got it wet again, then found out I had two uniforms, one wet and one dirty.

In keeping with the spirit of the game, there's no ordinary reply when you tell the computer to get something. Instead you get a sharp "Right, me old chum." Should any copies of the game reach America, it will only reinforce their beliefs that we still talk like Dick Van Dyke.

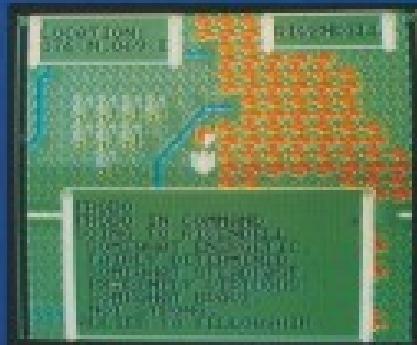
The authors have put a nice line in subtlety humour, which I greatly appreciate. What a shame about the programming.

If you can ignore the niggles that I picked on, get down the frog and read us" give your dash to the greater in the sky. But if you're the sort who likes to be able to use single key inputs, logical geography and no bugs, knock it on the head.

One last question, Where was Kylie?

#### Touchline

Title: Touchline. Supplier: Top Ten Hits, 101, 12 Glades Enterprise Centre, Station Road, Thetford, Norfolk NR7 0AA. Tel: (0603) 607 600. Machine: C64. Price: £14.99 (Disk), £9.99 (Cass).



money and the scores will snap to the battle screen showing all the combatants, who will start the fight without your help.

You can affect the battle by using a cursor to move men into the fray or you can control one character directly. This may sound like a good idea but it doesn't quite work, as to select a character you have to position the cursor over his feet and the handles you take too long. This can be annoying, particularly if the battle has only a dozen action while the party moves through the arena.

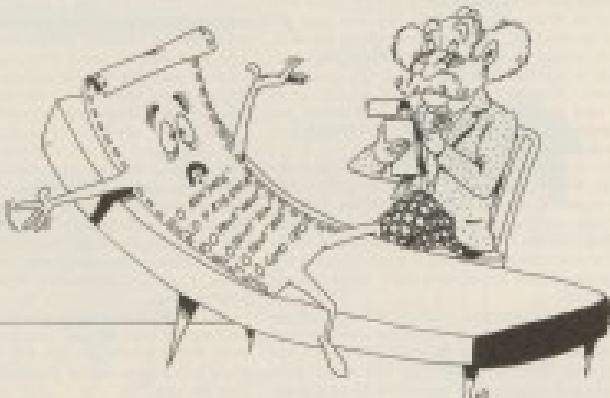
Unfortunately, this specimen of football management game which could have been so good.

#### Touchline

Title: War in Middle Earth. Supplier: Melbourne House, 24 Vernon Park, Petersfield GU12 2DN. Tel: 01273 6676. Machine: C64. Price: £14.99 (Disk), £9.99 (Cass).

# Program Analysis

*Programming can be made simple with these three C64 performance analysers*



## COMMODORE 64 PERFORMANCE ANALYSER

**B**ASIC is a programming language which makes it very easy for programmers to write complex programs, with a minimum of effort. We pay a price for this programming ease, and that price is often poor performance, that is, our Basic programs run slowly. Another problem which confronts the Basic programmer is what to do when a program runs without failing, but doesn't give you the results you expect. How do you find out what your program is doing without adding PRINT statements to your program to trace execution or interrupt execution at strategic points?

The Performance Analyser helps to overcome both problems. Not only does it map the logic flow in a Basic program, it also determines how long each Basic line took to execute. Thus the Performance Analyser is a generalised performance analysis tool for the Commodore 64.

### Performance Analyser TRACE Facility

Most commercial traces usually consist of a window displaying the first six line numbers on the screen at your Basic program rate. The line number scroll in the window as each line is executed, and the window may or may not interfere with your program output. You normally cannot trace a Basic program which uses line graphics, and you certainly cannot go back and check the line number sequence previously displayed. Although you can usually slow the trace display down (by the space bar for example), you have very little chance of writing down the line numbers on paper for a more detailed analysis.

The Performance Analyser overcomes all of these problems. It allows you to trace any Basic program which uses normal screen graphics, lines, screens, sprites or sound and does not interfere with the operation of the program. The Analyser will not slow

your program down, and allows you to give the trace display at your leisure. You may scroll backwards or forwards through the line numbers for as long as you wish.

### Performance Analysis

The Analyser also provides you with a tool to determine how efficient your Basic program is. When it displays the line number it also displays the time it took to execute the line. As you scroll through the line numbers you can tell at a glance which line numbers are slowing down execution and which line numbers are executed most often. Basic programs are the same as any other program - they follow the 80/20 rule. That is, 80 per cent of the work is usually done by 20 per cent of the program. The Performance Analyser is the tool you need to tell you which 20 per cent of your program is doing 80 per cent of the work, and how long it is taking to do it. You can then concentrate on making that part of your program more efficient.

## Analysing a Basic program

The Analyser is written entirely in Machine Language, and is designed to cause as little interference as possible with the traced programs. The Analyser is normally loaded at \$8912, and all Analyser variables and constants are contained in the ZR RAM (\$8912 to \$8959). Your Basic program does not use the RAM between \$8940 and \$8951, any low storage locations it requires and the free RAM at \$8952. Should you require the RAM at \$8912, then set the top of Basic pointer (\$A,26) to the last RAM location available to Basic, and the Analyser will use 2K of RAM before this address. For example, if the top of the basic pointer is set to \$2788, then the Analyser will load itself at \$8958.

Type in the Analyser loader program and save it as ANALYSER1. Make sure you verify that what you saved is correct. To use the Analyser, simply issue a load "ANALYSER1" after setting the top of BASIC pointer if necessary. ANALYSER1 will set the required Basic pointers, POKE the Analyser Machine Language logic into the correct RAM, relocate all required MLI addresses and print messages to indicate how to start and stop the Analyser and display the trace data. The following messages are displayed on the screen by ANALYSER1 during execution:

```
LOADING THE ANALYSER AT
$8912
LOAD OR
RELOCATION OR
1 START ANALYSER - SYS 38912
2 STOP ANALYSER - SYS 38913
3 DISPLAY DATA - SYS 38918
```

If the load fails, or the relocation of address fails, a message is issued and ANALYSER1 stops.

Obviously to start the Performance Analyser you SYS 38912 or to the address displayed by ANALYSER1. You can do this from a program or from direct mode. The message TRACE STARTED is displayed by the Analyser, unless you start it from a program. The message is not issued to ensure that the Analyser does not interfere with program messages or displays.

After the Analyser has been loaded, you then LOAD the Basic

program or programs you wish to analyse. The Analyser monitors execution of your program(s), and saves trace data in the trace data buffer for later display. If you only want to trace part of a basic program, you would do the following:

```
8080 REM START THE
ANALYSER
8080 SYS 38912
8020 FOR TH = $8912 TO $8951
8030 S = A$CODE(TH)
8040 Y = PCODS(TH)/C
8050 NEXT
8080 REM STOP THE ANALYSER
8070 SYS 38913
8080 REM DISPLAY TRACE
DATA
8090 SYS 38918
1000 END
```

After your Basic program has finished, or you stop it executing, you can stop the Analyser if you want to. However, you don't stop it to display the trace data. You may issue a *GO* to trace another program if you want to.

Obviously, to stop the Performance Analyser you SYS to 38913 or to the address displayed by ANALYSER1. You can do this from a program or from direct mode. The message TRACE STOPPED is displayed by the Analyser, unless you stop it from a program. Again the message is not issued to ensure that the Analyser does not interfere with program messages or displays.

Finally, you may display trace data at any time by entering SYS 38918 or SYS to the address displayed by ANALYSER1, and of course you may do this in direct mode or from a program. The message GO TRACE DATA is displayed by the Analyser if there is nothing to display. Again the message is not issued if you are under program control. This is to ensure that the Analyser does not interfere with program messages or displays.

If there is data to display the Analyser presents it in full-screen mode, that is a page of full screen data consisting of line numbers and line execution times is displayed and the Analyser MLI program waits for you to press one of the function keys; F1 terminates the display, F2 scrolls back to the previous page of data and F7 scrolls forward to the next page of data.

You may scroll back and forward through the trace data for as long as

you like with function keys F3 and F4. When the end of the trace data is found, the number of lines executed and the total execution time is displayed, and the Analyser MLI program waits for you to press a function key. The Analyser will only recognise F1, F2 and F7 function keys. All other keys are ignored. If you scroll forward from the end of the display, just wrap around to the start of the trace data again. You won't scroll back from the top of the trace data, you may only scroll forward.

**NOTE:** Trace data will be displayed automatically when the trace data buffer area is full. The trace data buffer is actually the RAM under the BASIC ROM. As much trace data as possible is stored there before the execution of the Basic program is interrupted and the trace data displayed. If you want your Basic program to continue, simply press F1 and the trace display is terminated. Your program begins execution from where it was interrupted. If you want to browse the trace data, then use F2 or F7 to scroll back and forward through the data.

## How the Performance Analyser Works

The Analyser works by monitoring the execution of Basic programs via the character display screen in low storage. As each program byte is interpreted, the Analyser checks to determine if the current line number (157,268) has changed from the previous byte read. When the line number changes, then the Analyser stores the line number and current time in the trace data buffer under the BASIC ROM. This is done until such time as the trace data buffer is full.

When the buffer is full, the Analyser saves the first 2K of low storage (\$0-2047), colour RAM and various control registers in the RAM under the KERNAL ROM. The trace data is then displayed, and when the display is stopped via function key F1, the Analyser restores the first 2K of low storage, the colour RAM and the various control registers. This allows the Basic program to restart execution from the point where it was interrupted, and the program screen is restored, as well as character colours and backgrounds.

If your Basic program uses the RAM under the Basic or KERNAL ROMs, then you cannot analyse it with this utility. Note also that if your

Basic program needs the time (TH = "000000"); then the Analyser will not fail, but the execution times displayed will be unpredictable.

## COMMODORE 64 PROGRAM ANALYSIS

**C**ommender 64 Program Analysis (COMPAANAL) is a Basic program which analyses the contents of any Basic program and displays the information on the screen or printer. COMPAANAL first displays summary information which contains the program name, the size of the program in bytes, the number of lines in the program, the total number of commands (e.g. PRINT's, GOTO's, IF's etc.) and the number of variables.

Once the summary data has been viewed, a detailed list of the commands used in the Basic program and the number of lines each command is used is displayed. When you have finished viewing the command data, a detailed list of the variables and the use of each variable is displayed, and when you have finished viewing the variable data you may end the display, ask for the information to be reprinted or end the data to your printer.

### Using COMPAANAL

COMPAANAL allows you to analyse your Basic programs. It does this by running in the 4K of free RAM at \$F010 to \$F047, and loading the Basic programs it analyses at \$0000. By not using the RAM between \$0048 and \$0063, COMPAANAL is capable of analysing the largest Basic programs. However, with only 4K of RAM to run in, COMPAANAL will not easily analyse large Basic programs because many garbage collections will be done to ensure that there is sufficient space for COMPAANAL to operate correctly. Also, only 50 variables can be displayed because of space constraints.

Obviously if COMPAANAL is to run in the RAM at \$F010 then some changes need to be made to Basic pointers in low storage. The start of Basic and end of Basic addresses need to be changed as well as the start of variables etc. These changes are handled by the COMPAANAL loader program, LOADER is the basic loader program which automatically loads

COMPAANAL. It sets the low storage pointers, and then uses the dynamic key facility to automatically load COMPAANAL.

You must create and save LOADER first on tape or disk. Next type in COMPAANAL and save it directly after LOADER on tape or on the same disk as COMPAANAL.

Note that if you are using disk you need to change line no 10 in LOADER from LOAD "COMPAANAL", 1,1 to LOAD "COMPAANAL", 1,1 so that COMPAANAL will be loaded from disk and not tape.

Once you have saved LOADER and COMPAANAL to tape or disk then simply load LOADER and RUN it. LOADER will set the various low storage pointers and then set up the screen and keyboard buffer so that when it ends, COMPAANAL is automatically loaded at \$F010. When COMPAANAL has been loaded it begins execution automatically, clears the screen and places the first message on the screen.

### LOAD FROM DISK (T/N)

If you want COMPAANAL to load the Basic program it analyses from disk, then reply Y. Otherwise reply N and the program will be loaded from tape. Before replying to this message, you should have the tape or disk which contains the program to be analysed in the cassette or disk drive.

The next message to be displayed is:

PROGRAM TO BE LOADED? - ?  
Your answer to this message tells COMPAANAL the name of the program it is to load from tape or disk to analyse.

COMPAANAL then uses the KERNEL load subroutine to load the Basic program at \$0000 and begin to analyse it. Since it may take some time to analyse large Basic programs, COMPAANAL places the line number being analysed in the top-left-hand corner of the screen while scanning the Basic program. When analysing is finished the summary report is displayed as follows:

—PROGRAM STATISTICS—  
PROGRAM NAME : COMPAANAL  
PROGRAM SIZE = xxxx  
NO OF LINES = xxxx  
NO OF COMMANDS = xxxx  
NO OF VARIABLES = xxx

### USE ANY KEY TO CONTINUE

You may view the summary report for as long as you wish. To move to the command report, simply use any key and the following display appears on screen:

#### - COMMANDS -

END	= 1
FOR	= 5
NEXT	= 6
DATA	= 10
INPUT	= 1
READ	= 1
GOTO	= 25
IF	= 30
GOSUB	= 37
RETURN	= 37
REM	= 8
POKE	= 5
PRINT	= 38
THEN	= 23
+	= 45
:	= 15
:	= 51
:	= 19
AND	= 1
^	= 55
MIDS	= 3

### USE ANY KEY TO CONTINUE

If all commands used in the program can be displayed on one screen, then when you press any key you will move to the VARIABLES display. If more commands are used than can be displayed on one screen, then the next screen of data will contain command data. When the last of the command data has been displayed and the USE ANY KEY message is displayed, when you press any key the list of variables appears on the screen. Note that - , / , < , > are considered commands when used in statements such as A=1+B\*C-D/E or IF X = 0 GOTO 1000.

When the commands are finished, the list of variables is displayed as shown:

I	= 2
X	= 4
RHS	= 5
Z	= 3
ZP%	= 4

### USE ANY KEY TO CONTINUE

When the last of the variables has been listed, COMPAANAL displays the following message:

R = RE-DISPLAY, X = END, P = PRINTER

If you press the R key, then all information beginning with the memory display is redrawn. If you press the X key then program execution is terminated and the final line message is displayed.

#### TIME TAKEN = XXXXX

This is the time in seconds it has taken C64PANAL to analyse your program. You may then use C64PANAL to analyse another Basic program. Press P and the information is sent to the printer.

### Applying C64PANAL

C64PANAL has many uses. You can find the size of your Basic program, the number of variables you use and the number of lines in your program. The number of lines is important, because each line in a Basic program carries an overhead of 4 bytes (2 bytes for a line address and 2 bytes for the line number). A 500 line program uses 1,000 bytes of storage for line addresses and line numbers. If you have an extensive number of lines, you can conserve space by reducing the number of lines (also known as condensing your program). You reduce the number of lines by placing multiple commands on the same line separated by colons, semicolons, blanks and entering REM commands.

Processing new lines also carries with it a performance penalty. The more lines in a Basic program, the longer it normally takes to run. By reducing the number of lines, you normally reduce program execution times. C64PANAL will tell you how successful you have been at reducing the number of lines in your program. It will give the size of your program and the number of lines before condensing, and then after you have made your changes you can run it again and get the new figures.

The detailed list of commands (PRINT#, GOTO#, IF etc.) can also be used to reduce program size and increase performance. For instance, if you find that you have a very large number of IF commands, then you may be able to reduce them by using the ON command. For example if you have:

IF CC = 1 GOTO 1000

IF CC = 2 GOTO 1100  
IF CC = 3 GOTO 1200  
IF CC = 4 GOTO 2000  
IF CC = 5 GOTO 2100  
IF CC = 6 GOTO 2000

then you could replace the IF commands with one:

ON CC GOTO 1000, 1100, 1200, 2000,  
2100, 2200

It is also interesting to see the pattern of commands in various programs and which commands are used most frequently. In string operations the LEFT\$, RIGHTS\$, MID\$ are well figure prominently. However, the most common commands used are the IF, GOTO, FOR and NEXT and PRINT.

The list of variables is a powerful tool to help in the creation of your Basic program. Basic maintains a list of variables, and the closer a variable is to the start of that list, the less time that is needed to find the variable when it is referenced in a statement. For example, every time IF X = 3.7 THEN 100 is executed, the X variable must be found in Basic's list of variables to check if it is 3.7 or not. Then the closer X is to the top of the list, the faster it is found. The order of variables makes a significant difference to the execution time of your program if you have a large number of them. C64PANAL helps by giving you a guide as to which variables ought to be defined first so that they appear at the top of Basic's list of variables. You can ensure the order of variables by defining them in the following order:

X = DA, Y = BY = DPC = QTRK = 0 etc.

X will come first, A second, Y third in the list and so on.

If you have Basic programs where execution time is crucial (for eg. games programs) then C64PANAL will be an important tool to help you analyse those programs and make them faster.

### COMMODORE 64 SWITCH

**C**ommodore 64 SWITCH is a short Machine Language (ML) program which resides in RAM just before the BASIC ROM. It occupies storage locations 40794 to 40999. C64 SWITCH allows you to partition your C64 into two logical machines. You switch between the two partitions or regions with a single key

stroke. With this utility, you can load two Basic programs at once and compare them or work on them. However, you cannot basic both programs running simultaneously.

### Using C64 SWITCH

C64 SWITCH allows you to set variable region sizes. The regions are designated area 00 and area 01 and region 0 will extend from location 2049 to the limit your set, while region 1 extends from the end of region 0 to 40793.

To use the switching function, simply load SWITCH, which is a Basic loader program. When you run it, SWITCH will load the ML routine at 40794 and display the message:

ENTER REGION = ENDING  
ADDR = >?

You enter the ending address for region 0 (and thus region 1 starting addr) and the final message is displayed:

REGIONS 0 AND 1 INITIALISED  
REGION ACTIVATED = 0

To switch between the two regions use the F1/F2 keys. F1 will activate region 0 while F2 will activate region 1. The active region is displayed in the upper right-hand corner of the screen in reverse video. To deactivate the SWITCH, simply hit RUN/STOP/RESTORE or turn the C64 off and on.

### Applying SWITCH

C64 SWITCH has three main uses. You can load two Basic programs at once, and work on them or compare them. You can use region 1 as a data region which is accessed by a program in region 0. SWITCH was originally written for this purpose. Finally, you can use SWITCH as a means of merging two programs. If you want to add code to a program in region 0 from a program in region 1, simply LIST the statements in region 1 on the screen, press F1 to activate region 0 and then move the cursor over the lines you want added and press RETURN. Each line will be inserted into the program in region 0.



*Enhance your help functions with this handy utility*

*By Mark Everingham*

In the old Commodore advertisement you know, the four-page epic which managed to link Charles Babbage, an elephant, and a tidy-bean called BJ to buying a Commodore computer for Christmas, special emphasis was laid on the 'HELP' function of the C16 and Plus/4 computers. Commodore claimed it helps you to debug your programs, yet I have owned a Plus/4 for several months now, and can honestly say I have never used the HELP function, except for the novelty. That BBC Micro has a command \*HELP which lets the user of a given command on a sideways ROM, and I decided to implement such a function on the Plus/4. I decided on these features it should have:

1. It should be compatible with the Commodore C16.
2. It should not take any memory from the program.
3. It should not interfere with the normal HELP function.

A tall order? Well, I decided that to allow a reasonable amount of help on the C16, the program must use the Disk Drive. That way, I could put it in the cassette buffer so as not to use up any memory, and a CHROUT wedge like the DOS SUPPORT program seemed appropriate to allow for the normal HELP function. The result is a 142 Byte piece of machine code using standard PRG files on a Commodore Disk Drive such as the 1581.

### The Programs

Listing 1 is a short and sweet Basic Editor I wrote to produce HELP screens and store them on Disk.

Listing 2 is the Basic Loader program for the HELP routine. It Pokes all the code into memory and defines the function keys.

### Listing 1 - The Screen Editor

When you've entered and defined the program, save it onto a disk using DSAVE 'HELP EDITOR' and RUN the program again. You should be presented with a blank screen showing the usual flashing cursor in the top-left corner. At this stage, the editor acts just as if you were editing a document - type text, in normal or reverse, graphics symbols, anything you want, even use the ESC functions to format your screen.

Pressing RETURN from the first level puts you into the command mode. A bar will appear at the bottom of the screen with three options. Press the relevant function key to select each one.

LOAD prompts for a filename and attempts to load this HELP screen from the disk.

SAVE prompts for a filename and saves the current screen to disk under that name.

CONT puts you back into level 1 edit mode.

When you've created a help screen, save it onto Disk under a suitable and memorable name, and exit the program. The HELP command now has something to work with. If you DIRECTORY the disk, you will see a file 'filename.H'. The 'H' designates a HELP file.

### Listing 2 - The Basic Loader

Now that you have some data on disk type in Listing 2, the Basic Loader, and save it on disk using DSAVE 'HELP PROGRAM'. When you have run it, try pressing function key 1. You should see the message 'SYS 932REM ON'. SYS 932 turns the HELP command on, press function key 2 or type SYS 930 to turn it off.

With the HELP command on, try typing in HELP (RETURN). This



simply does the normal HELP function - if your program has no errors in it, nothing should happen. Now, try typing: HELP 'Filename', where Filename is the name you saved your Help screen under. If all is well, your screen should load and the READY prompt will appear at the bottom of the screen. If this does not happen, something is wrong - if you get an error message "STRING TOO LONG ERROR" this means that you have tried to type in a name more than 14 characters long. If you get the error message "DIRECT MODE ONLY ERROR" then you have tried to use the new HELP command from within a program. If neither of these, use PRINT D53 to find the error. The Syntax, and errors returned by the HELP command are shown below:

### Syntax of the HELP Command

HELP [RETURN] - Normal HELP function

HELP (Filename) - Loads help file called 'Filename' from disk  
SYS 40 - Turns HELP Command On  
SYS 80 - Turns HELP Command Off

### Errors returned by the HELP Command

All DOS Errors - a fault concerning the disk

STRING TOO LONG - You have typed a filename longer than 14 characters

DIRECT MODE ONLY - You have tried to use HELP in a program

### Information on the HELP Program

The HELP Command resides by default in the first 147 bytes of the cassette buffer (K12-32F). Note that when it is installed in this area of memory, the program will be erased by pressing 'RESET'. I placed it here to avoid clashes between the Plus/4

and the C16. However it may be relocated by changing the address A in line 9 of the BASIC Loader Program.

### Practical use of the HELP Command

When creating HELP files, try to give them memorable and unambiguous names. It is also a good idea to make them short, though the '\$' and '%' designation may be used in filenames. For instance, if you are going to replace the manual with a HELP disk, divide the files into commands, each showing the syntax and a few examples of the command's use. E.g. Typing HELP %CIRCLE% might bring up all the different ways of using the command CIRCLE. The thing of utmost importance is to "use your common sense"! In a system which is very powerful when used properly, but could end up not being helpful at all if help files are designed without thinking. Anyway, I hope it will be very useful to you!

# GAMES UPDATE

## Super Cycle



The budget becomes continuous with this classic bike racing game that topped the charts in its full-priced format. Myriad Challenges wait to test man and machine.

## The Deep

If you tried to imagine what a game called "The Deep" would be like, you'd probably think of barrel-wrecks, sharks, octopuses or perhaps submarines.

You certainly wouldn't imagine that you'd spend most of the game on the surface! Unfortunately, this is exactly what you do in this rather disappointing game.

I remember one of a very old arcade game which I think was called Submarine Attack, in which you controlled a submarine with depth charges that had to survive an assault of submarines that released mines and fatal torpedoes at you.

In The Deep, your task is to destroy subs, after which small bags are occasionally released and float to the surface, staying there for three seconds. If you manage to avoid the floating mines and reach them in time, a helicopter will fly overhead and drop a package that you must catch. These packages increase your chances in the game by recycling hydrogen speed to your ship so you can dash across the surface, smart bombs to wipe out all subs, censor missiles that can be aimed and destroy everything in their path, or vulnerable pods that can reflect the bolts that

in a battle against other riders and the clock, as you have to cross the line before the seconds run out to qualify for the next round.

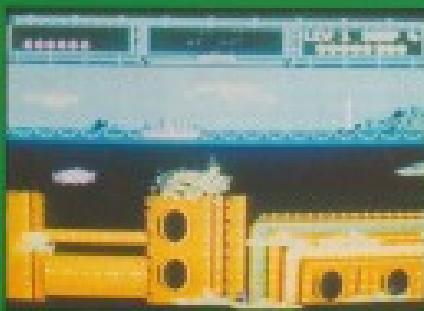
The early tracks are easy enough, and although the other riders can get in your way and even cause a crash, you'll easily catch up the time. Later on things get tricky, with the addition of slippery road surfaces to send you sliding, treacherous icy conditions, obstacles such as barricades and cones to send you crashing, and a higher-time limit where every second will count.

To add variety to the races, the scrolling landscape changes from day to night, country to city, and mountains to desert, the most impressive being the race near Cape Canaveral.

Avoid action at its best.

### Timelines

**Title:** Super Cycle. **Supplier:** Acorn (OS Gold). **Disk:** 2/3. **Media:** 3½". **Price:** £2.99.

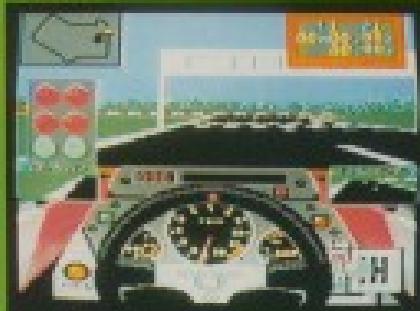


Be on the sea bed to clear a screen.

There are three timeline games in which you must destroy a giant ship, a massive submarine and protect a fleet of racing yachts by shooting down the missiles that are being fired at them, but then it's back to the main game again.

### Timelines

**Title:** The Deep. **Supplier:** US Gold. **Disk:** 2/3. **Media:** 3½". **Price:** £21.99.



## GRAND PRIX CIRCUIT

**D**espite the split between Electronic Arts and Accolade, the reign of Accolade imports continues from the EA stable. This latest one attempts to recreate the world of Formula 1 racing, and offers you the chance to drive for either the Ferrari, Williams or McLaren teams in a world championship against nine other drivers and over eight Grand Prix races.

Selecting the team you will drive for also selects the type of car you will drive — for example, the Ferrari is slower than the others, but is said as likely to spin and is a good car to start with. When you think you can control the car, you may want to swap to a Williams or McLaren. A more detailed appraisal of each car is presented on-screen in a display that shows the power curves, torque-power and torque-engine revs, gear box, chassis and weight, so that people who think these figures are important can use them to decide which car to use in the championship.

Perhaps most important is the game level you choose to play at, as this can determine whether it will be a quiet, sporting race without any mechanical problems, or a bitter fight where the best car and driver will win. The problem with the car is keeping it going at a speed fast enough to maintain your race position, but slow enough to stop it from blowing up.

Tires are also a major headache, particularly in the longer Grand Prix, where cornering too quickly and spinning can cause mass wear. They can be changed in the Pit, but this can cost you valuable seconds if your Pit stops aren't quick enough.

The first Grand Prix of the season is at Rio de Janeiro in Brazil, but before you can think about the world championship points for winning it, you have to drive round

the circuit in a qualifying lap. This not only gives you a preview of the track, but the time you take decides your position on the starting grid.

The race screen display shows your view of the track and your controls, which include a tachometer to watch for engine strain, a damage indicator that plots the condition of your car, especially braking and handling, a speedometer, start/stop button to check that corners are set at alignment, and a map-key that includes a line drawing of the course and a flashing dot to plot your position.

Starting can be a little tricky at first, as moving the joystick turns the wheel, and you must remember to turn it back again as it doesn't auto-steer, unlike most driving games. However, once you've mastered the basics you have to contend with the competition, particularly at the higher power levels when any collision will put you out of the race.

The corners are the best place to overtake, where the driver with the strategized move will take the lead — it's all down to when you apply the brakes. If you brake after your opponent, you'll go around the bend in the lead, but if you leave it too late you'll spin off.

While Grand Prix Circuit is a good simulation of a Formula 1 championship, a night race session will probably prove too much for all but dedicated Formula 1 fans. There is an option to race in a single Grand Prix, but there are better racing games for the casual driver. This one's for those who live on carbon monoxide car fumes.

### Specs:

Title: Grand Prix Racing. Supplier: Accolade (Electronic Arts). Length: Various Levels. Price: £39.99. Rating: 86%.



# Infiltrator



**H**ave's a second chance to fill the empty boots of the one and only Johnny "Jumbo Baby" McRobbins, as he flies off to the Wrecking Enterprise Chopper to save the world at least three times. The former full-priced game has been re-edited via the Kixx label, and it's excellent value for money.

His opponent in all this is the aptly described but unassisted Mad Leader, who is threatening all sorts of

destructive things if he isn't stopped, and Harbor Baby's the man for the job.

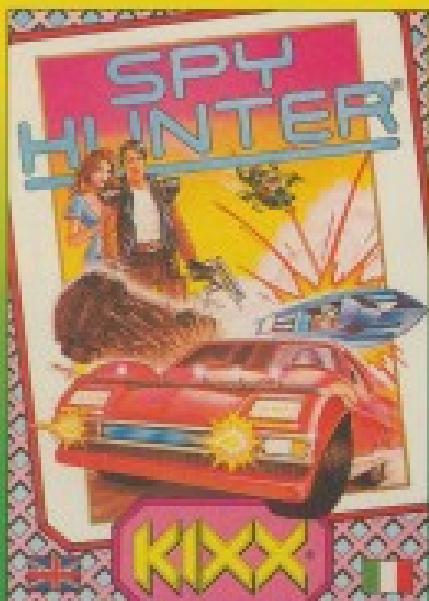
Each of the three missions follows a similar pattern - a three-stage game in which you must fly through various stages to find the Mad Leader's camp, then use tools to deactivate the security alarms and smash the buildings for secret plans, weapons and so on.

In fact, Infiltrator is three games-in-one, starting with a combat flight simulation in which you must bombard supply lorries to confuse the enemy, then lighting mills to draw them down. In the camp, you are armed with time bombs and sleeping gas to get into the camp without raising the alarm, break the buildings, things you sought as well as avoid the guards and smash (preferably Missionary-style) every object in every room. If the guards get too noisy, blowing their cover papers may send them off, if not, use some sleeping gas and then get out before the alarm sounds and raise the alarm.

Infiltrator was a smash hit in the States, but is well unheralded over here. At £2.99, it's a bargain.

#### Touchline:

**Infiltrator.** Supplier: Kixx (US Gold), 2980 2/2, Redhill Way, Redhill, Redbridge, RM7 2AX Tel: 0171 338 2388. Price: £2.99.



**T**his is the remake of the conversion of the original arcade game that began the craze for car shooters - a spin-off in a string of games including Road Blasters and Grid Driver. It all started in 1988 with Spy Hunter.

Naturally, it looks a little dated with its top-down view of a scrolling road network that you must patrol, but the gameplay is just as addictive, and has certainly stood the test of time.

The action begins as the weapons van pulls up at the side of the road and the Spy car rolls out into your joystick control and begins with the weapon gun as its only weapon. Your mission is to stay alive as long as possible, totally destroying and clearing the road of the villains - Road Lord, Barrel Dragger, Delancer, Switch Blader, and Mad Bone from the cemetery.

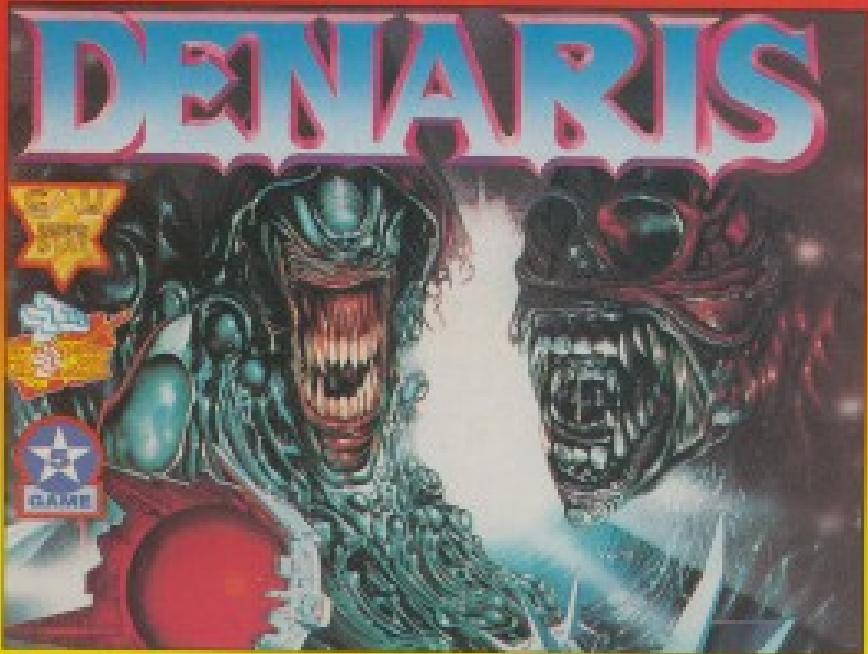
These villains attack you with appropriate weapons, which you must answer as best you can, while staying on course on a road that turns and splits to confuse you. At some stages the road disappears altogether, but luckily your Spy car then becomes a Spy boat and takes the battle on to the water.

Clearing enemies not only clears up the paths, but also earns you rendezvous with the weapons van and an upgrade for your weapons system, which includes rocket launchers, laser beams, homing missiles and rail-shots and much more to clear missions that get increasingly difficult.

Even after six years, Spy Hunter is still a good choice among.

#### Touchline:

**Spy Hunter.** Supplier: Kixx (US Gold), 2980 2/2, Redhill, Redhill Way, Redbridge, RM7 2AX Tel: 0171 338 2388. Price: £2.99.



**D**enaris is a planet with a problem. For years its citizens have been developing ever-advanced machines, and so it was almost inevitable that one day the machines would get so advanced that they wouldn't need the men any more.

By the time the Denarians realized what had happened, it was too late to launch a citizen assault. They tried anyway, but it just made the machines even more powerful. Their only chance to break the tyranny and escape from their underground prison is you, flying a small fighter.

The fighter is, of course, highly maneuverable, and can be improved by collecting items and debris from enemy ships. Before you start thinking that you've heard it all before, and that this is just another *Alone Again*, Denarist's attempt to tell you that it isn't. For one thing, two of you can play together.

The fighter is a DS-193 Eagle Fighter that begins the game unprepared for the onslaught ahead. You'll face a barrage of meteorites, waves of machine-armed Denarians, and the firepower of land-based gun emplacements. The other player controls a satellite that's accessed by players and collecting a crystal early in the game. From then on, it can either move independently or dock with the fighter in flight at what happens in the one-player game.

As with *Alone*, you can pick up objects to improve your ship, but there are limits to what they can do. They have specific effects on your ship, and come in the shape of crystals, extra weapons and tools. Crystals are split into four groups, with the Omega crystal activating and improving

the firepower of the satellite, and the Zeta stars increasing and decreasing the speed of the alien ships.

There are five types of bolts that add a cumulative effect to your ship - they include a red bolt to increase shooting power, green to add to the number of missiles, blue to activate a temporary shield, grey to add 1000 bonus points and yellow, which acts as a smart bomb destroying everything currently on the screen.

Extra weapon symbols are less common, and include weapons known as the Scatter Shot, Lightning Bolt Shot and Power Shot. You can also collect protection symbols that fly above and below your fighter and deflect incoming missiles. You will need all this equipment and protection if you are to survive the level, as well as the inevitable satellite machine that guards the way to the next screen.

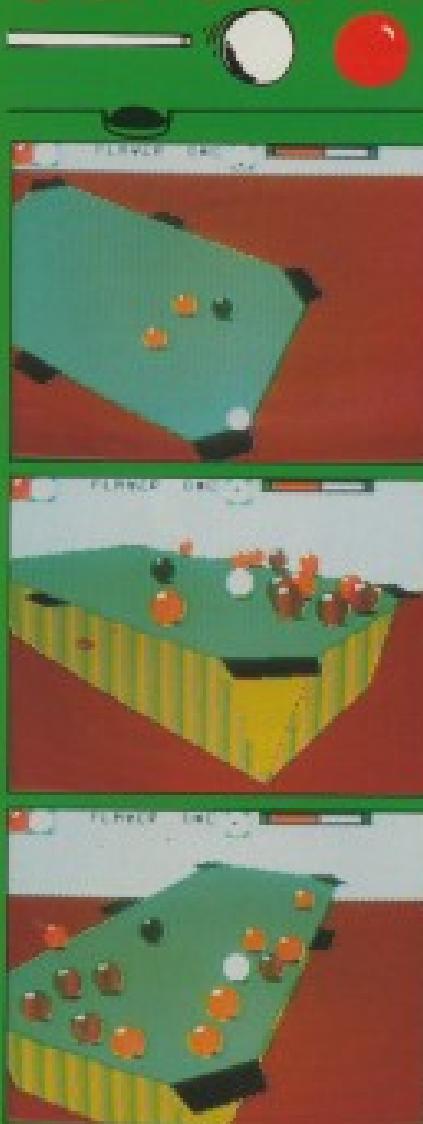
Perhaps the most important thing to learn is control over the fire button, as this is not a game where an over-firing profile is the way to get a high score. Instead you must choose the right moment to pull the fire button for a short burst of energy, or press and hold the button to build up the power shot that can take out a medium-sized asteroid or even a super machine.

Obviously, *Denaris* is inspired by games like *Alone*, but includes enough subtle differences to make it worth a second look.

#### POSSIBLE:

**Title:** *Denaris*. **Supplier:** *Peloton*. **Price:** £15.99. **Code:** 213. **Address:** 213 Waller Way, Moseley, Birmingham, B3 7AE. **Tel:** 071 555 2180. **Price:** £14.99.

# 3D Pool



Pool and snooker games are very popular, but they've always had the problem of being played entirely from a top-down perspective, making it difficult to shoot accurately. Now, thanks to Firebird and the skills of pool champion 'Mahesh' Jai Barker, you can get into the action in one of the best real 3D pool games.

It's quite a remarkable piece of programming — you can walk around the table, pull out for an overhead view and then get down close to aim your shot. There are no aiming lines or scores; it's up to you to get down close and judge the angles.

You'll need to practice a little on the controls before you can expect to clear the table in a single break. Left and right controls move you around the table, while up and down move in and out. Pressing the fire button up and down increases and decreases the power of the shot, starting at left and right controls. The horizontal position where the cue will strike the cueball. This decides the "cue" on the ball, and whether it will curve left or right after it's hit another ball or a cushion.

Cushion action is added by hitting the cue ball above or below its centre, and this, curiously enough, is controlled by the same controls that move you in and out. Therefore, to sink the last ball, you have to look down on the table from overhead.

Invariably, pool isn't just a case and be ready to take part in the tournaments against seven other players for the right to play Mahesh Jai himself. When you realize that he's pool champion of Steve Davis, you'll appreciate why you have to qualify first.

This is also the first DOS pool or snooker game which includes an actual tournament where each round is played over more than one frame. For example, you begin your challenge in the quarter final in a best-of-three match, then the best-of-five in the semi, and seven in the final.

The computer opponents provide a reasonable game, based on the fact that they know what they're doing, whereas pool is learning. As you get better, you'll get closer to the challenge match with Mahesh Jai. If that's too much for you, or you find it too easy, then you can opt to play against another human, or try one of the 10 trick shots with which you can amaze your friends.

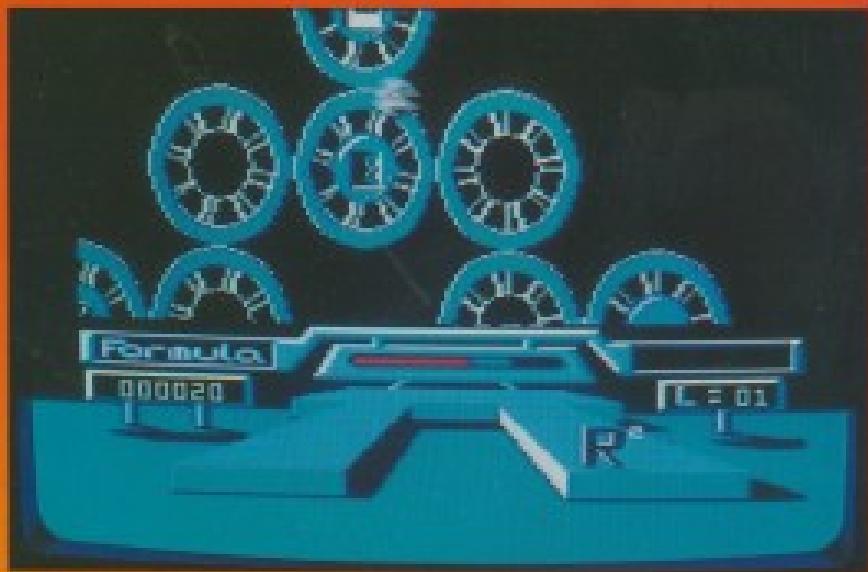
The game plays the popular path version of pool, in which you must pot all seven of your coloured balls before sinking the black to win the match. However, any foul there (and the computer opponents occasionally play what I would describe as deliberate misuse) are punished by sending the other player an extra shot and a free ball.

This game is definitely a cut above the usual 3D-style of pool game, but it does have some rough edges. Although the balls have shadows and reflections, they slide rather than roll across the table, but that's probably more to do with the technology at the hardware than the programming. The odder thing is the lack of cushion. Visually, they're there, but the balls seem to drift through them and hang off the edge of the table. However, I doubt there'll be too many complaints about these minor graphical panns.

As a first 3D pool game it is exceptional and free from the ridiculous seven cushion shots that plague many 3D versions.

#### Specifications:

**Table 3D Pool.** Supplier: Firebird, 64-76 New Oxford St., London, WC1A 1PS. Tel: 01 580 7347. Price: £29.95 (basic), £37.95 (plus).



# Bargain Bucket!

*Six new budget games of varying quality are given the once over by Gordon Hamlett*

**F**our titles this month, offering no different games, so everybody should be able to find something to suit both pocket and taste.

Moving Omega from Big Red Men you try to explore and deactivate an alien object - Omega. The problem is, you only have one hour of real time to accomplish this. There are four different reactors that need to be shut down in this period in order to ensure the safety of life, the universe and everything.

You start off by designing your own robot, each made up of four different components - base, weapon, sensor and power. You must then explore Omega, controlling your robot either manually, on automatic or by programming it.

This is an intriguing game, but it's let down somewhat by an inadequate set of instructions so that even after prolonged playing, I still had little idea of what exactly I was trying to accomplish.



Three games for less than a pound each in the box set of 22 from Silverbird. First of these titles is the decidedly *Bug 'n' Bounce*, your chance to try out such sports as flying, body power and acrobatics, set in the likes of Victoria Station and Geopress Ubog. Bouncing off the ropes to build up speed, you should attempt to collect your opponents in a bid before attempting to get him to the canvas.



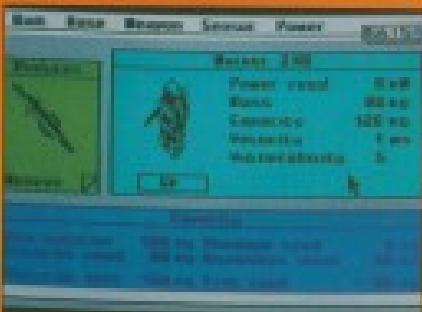
The other two games are somewhat less interesting. *Bug Body* has you trying to qualify for the next race by finishing in the top three in your current one. It is not sufficient just to complete the course; you have to perform stunts and tricks as well. Collected coins help to boost your energy and wheels to repair the damage caused by collision with the other players.

The final game is *Ninja Master*, which is a very clever but unusual action game. But one so bad that it's almost worth buying for that fact alone. Dodge arrows, break chop logs, leap off branch stumps with your sword and finally shoot down enemies with your blowpipe. If you've muster enough courage.

*Math Squared*, also from Big Byte, is a strange game. You must chase round bugs trying to pick up the different elements of assorted mathematical formulae. By doing so you round the perimeter of the bugs and the shapes. Contact with these decreases your IQ, although you can restore the by collecting books. Other objects may be beneficial or hazardous to your journey. I didn't particularly enjoy this game first time round, and I've added that final for *not*ナル多見。

The final game this month, and by far the best of the bunch, is *Over Core II* from Mastertronic's Ricochet label. Perhaps the most notable aspect of this sequel is that it is actually different from the original - something most unusual in this industry. The evil Melon has developed a race of Superoids, and it's up to Dan to penetrate the four levels of the Melon's ship and sabotage the control bays.

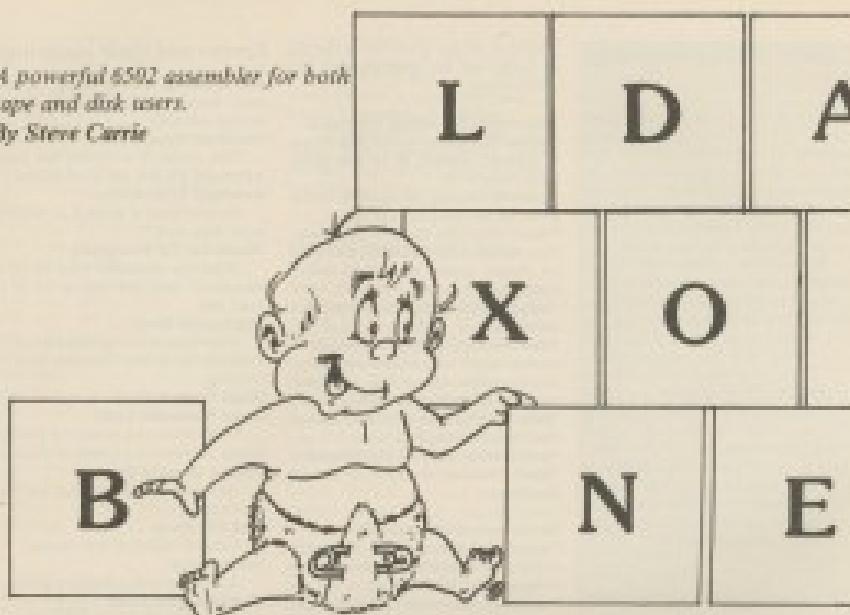
You can also choose to play the Melon, attempting to invade the Superoids' own outer space. In either case, you only have a limited amount of time to accomplish your task before moving onto the next level. This is a good-looking game that plays superbly well, and if you don't already have it in your collection, I suggest that you go out and pick up a copy straight away.



# Bargain Bucket!

*A powerful 6502 assembler for both tape and disk users.*

By Steve Carrie



# The ASM Assembler

The ASM assembler is a dual mode system whereby 6502 assembly language programs may be compiled from either one or more disk files and/or memory. It provides a set of commands to control its various functions, and also allows output to a Commodore printer.

The system has two modes of operation, disk mode and memory mode. In disk mode, the source code file is read from the source disk drive, and its compiled output sent to another file on the destination disk drive. In memory mode, the source code is read from memory (where it is edited in editor mode) and output to memory. Code relocation facilities exist to allow a program to be assembled to run at one address but placed at another.

The ASM system has two error modes, fatal and nonfatal. In fatal mode, an error will cause the assembler to halt, whereas in nonfatal mode, the assembler will process the whole file, leaving errors as it goes. Some facilities also exist for control of disk drives and errors reporting from these devices. A positive relocation facility allows all output which is normally sent to the screen to be sent to the printer at device numbers 4 or 5. Source and destination drive numbers may be set before assembly commences. Whilst using the editor, any Basic direct mode command may be issued.

Whilst ASM was designed primarily for use with disk drives, the fact that it supports memory assembly allows tape users to make use of it. The memory mode was originally

designed to allow short routines to be tested without having to resort to disk usage. Even in memory mode, you may still make use of files from disk as the file-link/file-include facility still works.

## Getting it all in

Listed here as a Basic loader program, ASM represents a considerable typing task. You may type in the program directly as it stands, but don't run it until you have read the next bit (before running, assuming you have saved a copy to tape/disk first) execute these three direct BASIC commands first. These will set the memory configuration correctly.

## POKE 49, 0! POKE 44, 15! POKE 9999, NEW

Now reload the Basic loader and run it. ASM will be POKE'd into memory at the correct address and saved to whichever device you set the device no. to (1 or 10/10/11 on the listing). Reset the machine and load and run ASM. You should get a signature message and a flashing cursor.

When ASM is loaded and runs, it installs a small wedge into the BASIC system. This has two important effects:

- (1) Edited program lines are no longer tokenised by BASIC. This means that you cannot edit a BASIC program. This is similar to the EDIT program in my 'Constructing a Computer' series in a previous issue of *Your Computer*.

- (2) A set of additional commands are introduced via the special character \*. These commands allow you to easily access the facilities provided.

## The additional commands

The extra commands are as follows:

### Assembly

\*source Start the assembler in the mode set by the mode command \*load and \*disk. The operation is as follows. In disk mode, you are asked for a filename whose default extension is .asm. The output file will have extension .asm. In memory mode, the source code is expected from memory and output is to memory. During assembly, output of messages, listings, etc to the list device will follow the mode set by commands \*printon and \*printoff whilst the treatment of errors will be defined by \*fatal and \*nonfatal. Assembly may be halted at any time by pressing the RUN/STOP key.

### Assembler control commands

\*disk Sets assembler disk mode. Source code is expected from disk and output is to a disk file (see \*format). \*mem Sets assembler memory mode. Source code is expected in memory and output is to memory. \*fatal If an error is encountered during assembly, the assembler will stop. \*nonfatal If an error is encountered during assembly, it is displayed but assembly continues with the next line of source code. Note that output is still

produced in this mode but it should be noted that the program may not run correctly.

### Information and editing commands

- \*remove Removes a source program in memory starting at 10 and going up in steps of 10.
- \*symbol: Display the symbols being from the last assembly operation.
- \*show: Shows the mode of operation. This will show the assembly, error and list device modes as well as device selection.
- \*help: Lists all the available commands.
- \*reset: Sets the ASM status to default startup mode. All values are set to default status i.e. printer off, list error mode, memory assembly.
- \*info: Display assembly information from last assembly operation.
- \*load: Display source device directory.
- \*load: Display destination device directory.

### Device control commands

- \*source Sets source device. Argument is the device number which must be in range 1 to 11.
  - \*dest: Sets destination device. Argument is the device number which must be in range 1 to 11.
  - \*load: Sends a command to source device. The command must be in quotes e.g. "load <filename>" will format a disk.
  - \*disk: Same as \*load but for the destination device.
  - \*errr: Display device error for source device.
  - \*errd: Display device error for destination device.
  - \*printon: Enables printer output. Argument is the select code for the list device (0 or 1).
  - \*printoff: Disables printer output.
- Note that certain functions such as listing are contained from within a source program; e.g. sym, list (see directives).

### Available Operators

- The following are valid in an expression:
- Hex value, e.g. #aa\$C000
  - Ascii value, must be two e.g. #ia\$A "A"
  - Low byte, e.g. lba \$c symbol
  - High byte, e.g. hby \$c symbol
  - Addition e.g. symbol+exp another2
  - Subtraction e.g. symbol-exp another3

## Errors and their meanings

The following is a list of the various error messages which may be printed during assembly.

### Undefined Symbol Error:

This occurs if a symbol has been referenced but has not been defined.

### Redefined Symbol Error:

Occurs when a symbol is defined more than once.

### Identifier Not Recognised:

What the assembler thinks to be a mnemonic does not appear to be a valid one.

### Bad Symbol Error:

Something is wrong with a symbol. Typically an invalid character or it is too long.

### Illegal Operand Field:

### Illegal Mnemonic Field:

These two errors point to a general syntax problem in a source code line.

### Missing Operated Error:

An operand was expected but was not found.

### Disk File Error:

General failure of disk system.

### System Errors:

A problem with a directive is likely.

### Illegal Quantity Error:

Some overrange condition has occurred, typically a 16-bit value in a byte mode instruction.

### Illegal Addressing Mode:

An instruction was used in an incorrect way.

### Not X Or Y Index:

Only X and Y index registers are valid.

### Symbol Table Full:

Pretty fatal one this. It indicates that the space set aside for symbols has been exceeded.

### Branch Range Error:

Branch instructions are relative and may only operate within a certain range.

### Unfile Name Length Error:

The argument to an file directive is too long.

### Unfile Name Missing:

The argument isn't there at all!

### Bad Directive In Memory Mode:

You have used some directive not valid in memory mode.

### Bad Directive In Disk Mode:

You have used some directive not valid in disk mode.

### Cannot Open Another Unfile:

Trapping error message when you try to link another file whilst already linked.

### No Such Select Code For This Device:

You have tried to assign a device

```

10 : Example 0
20 : Reserve Mode
30 :
40 : org $C000
50 : ldat
60 :
70 : index wba $B0
80 : char wba, index=0
90 : screen wba $D400
100 :
110 : lda #Cscreen
120 : ldy #Cscreen
130 : sta index
140 : sta index=0
150 : ldy #0
160 : ldy #0
170 : lda char
180 : ldi $11 sta index=0
190 : decr
200 : bne fall
210 : inc index+1
220 : decr
230 : bne fall
240 : rts
250 :
260 :
270 :
280 :
290 :
300 :
310 :
320 :
330 :
340 :
350 :
360 :
370 :
380 :
390 :
400 :
410 :
420 :
430 :
440 :
450 :
460 :
470 :
480 :
490 :
500 :
510 :
520 :
530 :
540 :
550 :
560 :
570 :
580 :
590 :
600 :
610 :
620 :
630 :
640 :
650 :
660 :
670 :
680 :
690 :
700 :
710 :
720 :
730 :
740 :
750 :
760 :
770 :
780 :
790 :
800 :
810 :
820 :
830 :
840 :
850 :
860 :
870 :
880 :
890 :
900 :
910 :
920 :
930 :
940 :
950 :
960 :
970 :
980 :
990 :

```

```

10 : Example 0
20 : Uses ROM file
30 : STARASM
30 : now in memory or
31 : disk mode
40 :
50 : org $C000
60 : ldat
70 :
80 : lda "Example"
90 :
100 : lda $15
110 : ldy #0
120 : sta wba+$B0
130 : sta wba+$B0
140 : rts
150 :

Pseudo:

```

```

Type This In And Save To Disk
10 : Module "V68K.ASM"
20 : Create For Example 0
30 :
40 : ldat sta $C000
50 :

```

code other than 4 or 5 for a printer ("Printer") or a code rather than 8, 9, 10 or 11 for a disk drive ("Power, "Disk). Occurs in editor mode only.

#### Device Communication Failure:

Communication to a disk drive failed. May indicate wrong device number. Equivalent to Basic's "Device Not Present" error.

### Things to look out for...

When ASM is installed and running, the following information is relevant. The Basic chapter code is distorted to a new routine within the ASM code to allow the inclusion of the new commands. The program loads like a Basic program into memory starting at address \$0001. When it has run, the start of Basic is shifted up to about \$2500. You may still type any Basic direct command such as LOAD, SAVE, POKE, etc but creation is advised using POKE on addresses between \$0001 and \$2500.

In both memory and disk modes, code is edited above about \$2500. In memory mode, the system table begins in memory after the source program. This also applies in disk mode hence any program in memory will be preserved. This means that you should type "new" before commencing a disk mode assembly to maximize symbol space. During assembly, the BASIC ROM is excluded from the space from \$C000 to \$CFFF is left free. Symbols thus occupy the space from the end of any program in memory up to \$CFFF.

ASM should execute peacefully with the Basic interpreter. The "test" command may be used to resolve certain situations where the system is not operating correctly. However, it has a limited effect, and it may become necessary to precautions should the system still operate incorrectly.

ASM is source code compatible with my earlier PCL system assemblies published in a previous Four Commands issue and also my PLUSS assemblies. ASM's facilities are inferior, a reversion of the PCL assembler's facilities and ASM could therefore replace the PCL assembler if desired.

To help you become familiar with the system, I have included some example source files listing which may be assembled using ASM. The comment fields at the beginning indicate which mode they should be run in. Good luck!

### DMA Assembler Directives

**BYT** Byte value directive. Single values or strings in single quotes allowed.  
e.g. byt \$2, \$A, \$FF, "abcd", "

**WORD** Word directive  
e.g. word \$C000, \$D000, \$FFFF

**EQZ** Zero page equate. Used to assign a temporary value to a symbol.  
e.g. pointer equ \$PP

**EQA** Absolute equate. Used to assign an absolute value to a symbol.  
e.g. var equ \$2000

**ORG** Set code origin. In disk mode - also sets code load address.  
e.g. org \$C000

**RES** Reserve memory.  
e.g. res 80 (reserves 80 bytes)

**LST** Causes assembler to list during pass 1.

**SYM** Causes assembler to display symbols upon completion of the assembly.

**LNK** Chain to another file. When file has been assembled, the current file resumes assembling.  
e.g. llnk "symbolic.asm"

**REL** Relocation offset. The code origin is set by the org directive. This directive allows you to assemble code in two different address (org) while being used to assemble memory area (rel).  
e.g. org \$0000  
rel \$C000

# Disk Edit

*Drive further into your disks with the help of this article.*

*By Fergal Moane*

Disk editing is what separates a casual disk user from a professional. Once you can edit data, a whole world of seemingly impossible tasks becomes possible. Files can be locked, unstructured, closed, relocated, and renamed when you have the commands and the know-how.

Finally, a word of warning: don't edit a disk with important programs on it, unless you know what you're doing. Use unneeded disks for practice, and take backups of valuable disks. A good Disk Editor will make things a lot easier; you have no need for complex commands. There is a good example in the December 1987 edition of *Your Computer*. This is not essential though - you can make do with the commands and DISPLAY TABS on the demo disk you get with your drive.

## Commands

The commands regarding direct disk access are called the Block commands (a Block is another name for a Sector).



Your disk drive manual contains more detailed explanations, but a summary follows.

To use these commands, you'll need to have two files open - one for commands and the other is a buffer for data. The command channel you will probably be familiar with:

#### OPEN[RA]S[2]

The data channel can be any other number, but 2 or 3 are usually used:

#### OPEN[A][R][W]

After these open commands, PRINT 15 will send commands, and PRINT 5 will send data to the channel.

Note that when 'Drive' is mentioned, this means 0 for a single drive. The device is usually 0, but can be changed. See the examples on the disk for more information.

#### Block-Read

**SYNTAX:** PRINT #15, "B-R";  
channel;  
drive;  
track;  
sector;

This command transfers the required sector into the data channel (in our case 5). There are the GET 5 command to read the information into a variable.

It's important to note that Block-Read will only read up as far as the Block-Pointer, which is usually 0. The USERT1 command is usually used, as this sets the pointer to 255 automatically, allowing the sector to be read in one operation.

#### USER1

**SYNTAX:** PRINT #15, "U1";  
channel;  
drive;  
track;  
sector;

#### Block-Write

**SYNTAX:** PRINT #15, "B-W";  
channel;  
drive;

track;  
sector;

To use this command, fill up the channel with information to write, using PRINT 5, then use the command to write to the required sector. This is the exact opposite of Block-Read, so again USERT2 is usually used.

#### USERT2

**SYNTAX:** PRINT #15, "U2";  
channel;  
drive;  
track;  
sector;

#### Block-Pointer

**SYNTAX:** PRINT #15, "B-P";  
channel;  
location;

By using this command, you can specify where exactly in the sector you want the read/read or write to begin. This allows you to read or alter individual bytes in a sector, starting at 'location'. See the Disk Name program for a demo.

#### Block-Allocate

**SYNTAX:** PRINT #15, "B-A"; drive;  
track;

sector;

This allocates a bit in the Block Availability Map to show a sector is in use. It is used in conjunction with random access channels.

#### Block-Free

**SYNTAX:** PRINT #15, "B-F"; drive;  
track;  
sector;

This is the opposite of Block-Allocate, and frees up sectors for use without destroying the actual data on them. If a sector reads, the data will probably be overwritten, as the RAM has marked the sector as empty.

#### Disk Maps

Before you can use these commands, you will need some information on disk structure. This maps will provide this information, and information on file structure.

#### Editing

There are a number of files provided here for demonstration purposes. The best way to learn is to study these programs with the maps closest hand. They are heavily REMed, but here are some notes explaining what's going on. Even if you don't know anything, they are useful utilities to have.

#### Protect File

This program 'locks' (or presents a lock being scratched) the first file on a disk by setting bit 0 of the file type to 1, effectively ORing it with \$C0. This prevents accidental erasure, and has a < .sav> file name in the directory. By adding 12 to the last pointer number, and changing the sector number, any program in the directory may be protected.

#### Disk Name

This allows you to change the name of the disk without erasing the contents. It makes use of the fact that the disk name is stored at bytes 64, track 15, sector 0.

#### Load Address

This changes the load address of any program to a given address. It scratches the first sector of storage, and bytes 2-10 contain the load address. It is most useful with split data.

#### Unscratches

On scratching a file, the filetype in the directory is usually marked as being deleted. This program searches the disk for a scratched program and restores the filetype, resurrecting the file. You are advised to save the unscratched program to another disk in case of another accident. Note that this will probably not work if something has been saved to the disk since the SCRATCH, as it may have been saved over the old program. Enjoy your disk editing!

## BLOCK DISTRIBUTION BY TRACK

Track number	Block range	Total
1 to 17	0 to 20	21
18 to 24	0 to 18	19
25 to 30	0 to 17	18
31 to 35	0 to 16	17

## 1540/1541 BAM FORMAT

Track 18, Sector 0.		
BYTE	CONTENTS	DEFINITION
0,1	58,01	Track and block of first directory block.
2	65	ASCII character A indicating 4040 format.
3	0	Null flag for future DOS use.
4-143		Bit map of available blocks for tracks 1-35.
		*1 = available block 0 = block not available (each bit represents one block)

## 1540/1541 DIRECTORY HEADER

Track 18, Sector 0.		
BYTE	CONTENTS	DEFINITION
144-161		Disk name with shifted spaces.
162-163	160	Disk ID.
164		Shifted space
165-166	50,65	ASCII representation for 2A which is DOS version and format type.
166-167	160	Shifted space
177-255	0	Nulls, not used.
Note: ASCII characters may appear in locations 180 thru 191 on some diskettes.		

## SEQUENTIAL FORMAT

BYTE	DEFINITION
0-1	Track and block of next sequential data block.
2-256	255 bytes of data with carriage return as record terminator.

## PROGRAM FILE FORMAT

BYTE	DEFINITION
0-1	Track and block of next block in program file.
2-256	256 bytes of program information in C64 memory format (with key words tokenized). End of file is marked by three zero bytes.

## RELATIVE FILE FORMAT

DATA BLOCK	
BYTE	DEFINITION
0-1	Track and block of next data block.
2-256	254 bytes of data. Empty records contain FF (all binary ones) in the first byte followed by 00 (binary all zeros) to the end of the record. Partially filled records are padded with nulls (00).
SIDE SECTOR BLOCK	
BYTE	DEFINITION
0-1	Track and block of next side sector block.
2	Side sector number (0-5).
3	Record length.
4-5	Track and block of first side sector (number 0).
6-7	Track and block of second side sector (number 1).
8-9	Track and block of third side sector (number 2).
10-11	Track and block of fourth side sector (number 3).
12-13	Track and block of fifth side sector (number 4).
14-15	Track and block of sixth side sector (number 5).
16-256	Track and block pointers to 120 data blocks.



**WIN WITH  
COURSEMASTER**  
**THE COMPUTER HORSERACING  
PROGRAMME**

- BETTER AND FAIRER IN SECONDS - WIN DAILY NEWSPAPER TO ALL YOU NEED
- MAINTAIN YOUR BETS - Bookie's Bet and Pari. Free data entry
- AMAZING ACCURACY! - how you Can Beat The Odds
- Win on the unique portfolio of **PAST MORTGAGE BETS** now!
- CHESS
- Classify the best selection in every race plus these FREE FEATURES:
- First, Second and Third-prizes shown by forecasts and fractions etc.
- Instantaneous odds calculations of race
- Authority works on your maintained account portfolio including
- THOROUGHS AND DUSKIES, WIN AND NOT WIN, PATENTS, VANDALS, CARAVANERS, HATRED, HILL, BOSS RATCHET and LONG CLOTH from Chester's advice
- WIN-PRIZE and your BETTING SLIP for you
- Maintains a BANK ACCOUNT - BET Like PROFESSIONAL & Get Rewards on your bets in any number of accounts. Keep a complete record of your betting as complete COMPILER REPORT against your bank account
- PAYOUT - THE AMAZING COURSEMMASTER SYSTEM! This payout system is included in the programme. A system which payout's your money based on your odds and losses. It's interest free for all your money won on the day!
- Kept up with 10 years BETTING RECORDS and MARSHAL

**FREE HOT TIP OF THE MONTH SENT TO YOUR PRINTER**  
Subscriptions: £15.00 per year  
All Books £1.50. **Commodore 64/128**:  
Amstrad CPC6128/664/6100,  
PC, Apple II, BBC B+,  
Acorn Atom, Amicas, Amiga,  
Amstrad CPC1640, BBC B+ & Master, Sinclair,  
Amstrad CPC, All Amstrad PCs  
etc.

Please state machine and disk type

**INTERSET LTD. (DEPT BC), FREIGHT POST (Pwllheli,  
Wales) 8 Gilpendale Close, Seven Cawr, Wrexham,  
Wales LL12 8BR. Or large SAE for  
further details.**

**PRINT ENHANCER Plus/4 disk ..... £19.99**

Print shading publishing. Print/Format enhanced. High quality graphics, from BASIC, C, B1 and C64 disk. Standard with 16 screens and bitmap printer programs. M65-8080, VIC 1080.

**R232Z INTERFACE 64, 128 or Plus/4\* ... £19.99**

Measuring only 10x10cm, the compact and robust unit connects Commodore to thousands of RS232C user-port standard. Compatible with BBC, Acorn, Spectrum, Amstrad, Amicas, Amiga, Apple, Radio Shack, printers, modems and other devices. Supplied with 2 meters of cable and the main menu is terminated with a main (menu) screen. Customisation of a custom menu made to your requirements. Our after sales service for unusual projects guaranteed to get you up and running. Suitable with a terminal emulation program, the interface and printer can be used together.

**IBM PC File Transfer Utility ..... £29.99**

Captures text or binary files via the serial RS232C port. Requires an 8 MHz 68000 based 486/386. Simple instructions provided by on-line help.

**VINUMDATA TERMINAL 64 or Plus/4 ... £14.99**

Accesses Parallel, Modem, Disk drives etc. Works with any user programmed interface and 1000+ drivers.

**CENTRONICS CABLE 64, 128 ..... £18.99**

Link Commodore, Amstrad, Amicas, Amiga, BBC, Radio Shack, or Z80 or 6502.

**6502 ASSEMBLER 64 or Plus/4 tape/disk ... £12.99**

A sophisticated two-pass symbolic assembler and link editor which supports reloc and reloc items. The assembler, editor, source and object code may all be present in memory simultaneously. Facilitating rapid and interactive code development. Very fast with Assembler-coded codes.

**Z80 EMULATOR/ASSEMBLER 64 disk ... £32.99**

A unique integrated Z80 development package. The emulator simulates Z80 source code into assembly Z80 which runs on the 64 at about one tenth the speed of a 2MHz Z80. The cross assembler generates hex or binary Z80 output files. The disk contains a powerful editor and example programs.

Please specify "options" (about £20.00) at order by phone.

Prices include VAT and P&P. Cheques, orders and £5.00

Add 10p up to 1 page for delivery.

**YORK ELECTRONIC RESEARCH**

The Fishergate Centre, Dept YO, 4 Fishergate, York YO1 4AB

Tel 0904 381121

# A SUBSCRIPTION IS ONLY A CALL AWAY...

What could be a better way of keeping up to date with the latest news and developments in the world of the Commodore range of computers, than by ordering a direct subscription to 'YOUR COMMODORE', delivered direct to your door each month.

We've now made ordering a subscription easier than ever before by calling our CREDIT CARD HOTLINE. Simply give us a call, quoting your credit card details and delivery address and we'll do the rest! Remember, a subscription delivered to any address in the UK is POST FREE, all overseas subscriptions include postage.

Subscription Rates: UK £15.00; Europe £25.25; Middle East £35.00; Far East £26.40; Rest of the World £28.00 or USA \$42.00. (Airmail Rates on Request).



**Telephone 0442 876661/4**  
Between usual office hours.

MONDAY	9.00
TUESDAY	9.00
WEDNESDAY	9.00
THURSDAY	9.00
FRIDAY	9.00

# Listings

#### Help Protection



中国科学院植物研究所

JONES, R. 1944-8, REED, R. P. 1978-1  
1980-48  
2000 PRINTERS' INDEX IN DATA... 1980  
2000 DATA 1980, P. 100, LINES 30-107, 189,  
190  
2000 DATA 8, 101, 200, 148, 18, 100, 189,  
190  
2000 DATA 100, 140, 204, 189, 12, 20, 209  
190  
2000 DATA 70, 201, 180, 189, 2, 179, 118, 1  
87, 10

PROVOST'S BULL PROCLAMATION

Unit 10



www.legis.psu.edu

#### REFERENCES



PHOTO BY GENE KELLY

```

72 DE POSITION-CLB SWN SET-LINE
73 DE POSITION-CLB SWN SET-LINE
74 DE POSITION-CLB SWN SET-LINE
75 DE POSITION-CLB SWN SET-LINE
76 DE POSITION-CLB SWN SET-LINE
77 DE POSITION-CLB SWN SET-LINE
78 DE POSITION-CLB SWN SET-LINE
79 DE POSITION-CLB SWN SET-LINE
80 DE POSITION-CLB SWN SET-LINE
81 DE POSITION-CLB SWN SET-LINE
82 DE POSITION-CLB SWN SET-LINE
83 DE POSITION-CLB SWN SET-LINE
84 DE POSITION-CLB SWN SET-LINE
85 DE POSITION-CLB SWN SET-LINE
86 DE POSITION-CLB SWN SET-LINE
87 DE POSITION-CLB SWN SET-LINE
88 DE POSITION-CLB SWN SET-LINE
89 DE POSITION-CLB SWN SET-LINE
90 DE POSITION-CLB SWN SET-LINE
91 DE POSITION-CLB SWN SET-LINE
92 DE POSITION-CLB SWN SET-LINE
93 DE POSITION-CLB SWN SET-LINE
94 DE POSITION-CLB SWN SET-LINE
95 DE POSITION-CLB SWN SET-LINE
96 DE POSITION-CLB SWN SET-LINE
97 DE POSITION-CLB SWN SET-LINE
98 DE POSITION-CLB SWN SET-LINE
99 DE POSITION-CLB SWN SET-LINE

```

## LISTINGS

MATING

## LISTINGS

Retourner



第10章 网络协议

www.english-test.net

```
47 449 PRINT#1,CLN,CDNC,FLL,OF  
48 5000,POKE32134,PEEK44014  
49 100000  
50 449 POKE2-38704,4271,POKE1,100  
51 449 POKE2-38704,4271,POKE1,100  
52 449 POKE2-38704,4271,POKE1,100  
53 449 POKE2-38704,4271,POKE1,100  
54 449 POKE2-38704,4271,POKE1,100  
55 449 POKE2-38704,4271,POKE1,100  
56 449 POKE2-38704,4271,POKE1,100  
57 449 POKE2-38704,4271,POKE1,100  
58 449 POKE2-38704,4271,POKE1,100  
59 449 POKE2-38704,4271,POKE1,100  
60 449 POKE2-38704,4271,POKE1,100  
61 449 POKE2-38704,4271,POKE1,100  
62 449 RETURN
```

PRINCIPLES - METHODS - PRACTICE

TOPICS • PERSPECTIVES

## LISTINGS

E3 311491 TPIKCI1M1347THB9-BKC1  
A81-709  
E3 311491 TPIKCI1M1347THB9-BKC1  
B8C0201149  
E1 311491 K1-A121114-17 : BKC-BK-09  
E1 311491 K1-BK  
E3 311491 K1-BK  
E3 311491 K1-BK

## Flow of Ideas



卷之三

180 900 BEST  
175 900 200-1000, REV. R. YELLOW,  
175000 1000000 "CHINESE"  
180 900 FORM-L7215 8000000 PAI  
  
175 900 ONE-ONE-CHINESE  
175 900 FORM-L7215 8000000 PAI

EF 800 PRINTERS, B&W  
EF 800 PRINTERS DOWN WHITE, B&W  
MACHINES FOR DRUGSTORES ETC, B&W  
EF 100 PRINTERS DOWN WHITE, B&W

RE 8121 177-19400  
RE 8122 177-19400  
RE 8123 177-19400  
RE 8124 177-19400  
RE 8125 177-19400  
RE 8126 177-19400  
RE 8127 177-19400  
RE 8128 177-19400

CD 1000 2000 3000 4000 5000 6000  
PC 8000 10000 12000 14000 16000 18000  
L0100 20000 30000 40000 50000 60000  
T7 1000 2000 3000 4000 5000 6000  
SF 1000 2000 3000 4000 5000 6000  
P 10000 20000 30000 40000 50000 60000  
PS 1000 2000 3000 4000 5000 6000

```
0017
0018    FILE POSITION
0019    100 00100000000000000000000000000000 DIRECTORY BLOCK
0020    0 00000000000000000000000000000000 00000000000000000000000000000000
0021    100 00100000000000000000000000000000 SPECIFIC-NAME
0022
0023    FILE POSITION
0024    100 00100000000000000000000000000000 DIRECTORY BLOCK
0025    0 00000000000000000000000000000000 00000000000000000000000000000000
0026    100 00100000000000000000000000000000 SPECIFIC-NAME
```

0.05%  
 50 TWO PRIMES "LUMINES-SPINE"  
 BACK SECTIONS INPUT-10-17  
 50 TWO PRIMES "LUMINES-B", LUMES, INPUT  
 17-18  
 50 TWO PRIMES "LUMINES-B", LUMES, INPUT  
 18-19

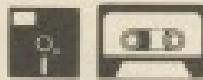
100 1950-1951 "LONDON, B.C., BY  
101 G. L. LEWIS IS IN THE AT THE  
102 JUNIOR HIGH SCHOOL, B.C."  
103 1950-1951 "VANCOUVER, B.C., BY  
104 G. L. LEWIS IS IN THE AT THE  
105 JUNIOR HIGH SCHOOL, B.C."

1000 PRESTONWOOD, DALLAS  
1010 972-248-1000; FAX 972-248-1011  
1020 1000 PRESTONWOOD, DALLAS, TX 75248  
1030 1000 PRESTONWOOD, DALLAS, TX 75248  
1040 1000 PRESTONWOOD, DALLAS, TX 75248

10. 100 PROBLEMS IN  
 20. 100 PROBLEMS, 2-3-4  
 11. 100 PROBLEMS, 5-6-7, 8-9  
 12. 100 PROBLEMS, 10-11-12  
 13. 100 PROBLEMS, 13-14

29.5-1  
290 PRINT "NUMBER IS: ",C7,1100  
400 FOR I=1 TO 1000 PRINT C7,  
500 READ RND(1)\*1000+1 PRINT RND(1)  
600 END

#### 4.2.4 Assembly



42 21 19647 19647 1,10  
43 21 19647 19647 1,10  
44 21 19647 19647 1,10  
45 21 19647 19647 1,10  
46 21 19647 19647 1,10  
47 21 19647 19647 1,10  
48 21 19647 19647 1,10  
49 21 19647 19647 1,10

DE 00 0000 11-17 FIGHT TEAM  
DE "TENNESSEE LINE" LA 100  
DE 00 0000-18  
DE 00 0000  
DE 00 0000-19  
DE 00 0000-20  
DE 00 0000-21  
DE 00 0000-22  
DE 00 0000-23  
DE 00 0000-24  
DE 00 0000-25  
DE 00 0000-26  
DE 00 0000-27  
DE 00 0000-28  
DE 00 0000-29  
DE 00 0000-30  
DE 00 0000-31  
DE 00 0000-32  
DE 00 0000-33  
DE 00 0000-34  
DE 00 0000-35  
DE 00 0000-36  
DE 00 0000-37  
DE 00 0000-38  
DE 00 0000-39  
DE 00 0000-40  
DE 00 0000-41  
DE 00 0000-42  
DE 00 0000-43  
DE 00 0000-44  
DE 00 0000-45  
DE 00 0000-46  
DE 00 0000-47  
DE 00 0000-48  
DE 00 0000-49  
DE 00 0000-50  
DE 00 0000-51  
DE 00 0000-52  
DE 00 0000-53  
DE 00 0000-54  
DE 00 0000-55  
DE 00 0000-56  
DE 00 0000-57  
DE 00 0000-58  
DE 00 0000-59  
DE 00 0000-60  
DE 00 0000-61  
DE 00 0000-62  
DE 00 0000-63  
DE 00 0000-64  
DE 00 0000-65  
DE 00 0000-66  
DE 00 0000-67  
DE 00 0000-68  
DE 00 0000-69  
DE 00 0000-70  
DE 00 0000-71  
DE 00 0000-72  
DE 00 0000-73  
DE 00 0000-74  
DE 00 0000-75  
DE 00 0000-76  
DE 00 0000-77  
DE 00 0000-78  
DE 00 0000-79  
DE 00 0000-80  
DE 00 0000-81  
DE 00 0000-82  
DE 00 0000-83  
DE 00 0000-84  
DE 00 0000-85  
DE 00 0000-86  
DE 00 0000-87  
DE 00 0000-88  
DE 00 0000-89  
DE 00 0000-90  
DE 00 0000-91  
DE 00 0000-92  
DE 00 0000-93  
DE 00 0000-94  
DE 00 0000-95  
DE 00 0000-96  
DE 00 0000-97  
DE 00 0000-98  
DE 00 0000-99  
DE 00 0000-100

12 12 12 12 12 12 12 12 12 12  
13 13 13 13 13 13 13 13 13 13  
14 14 14 14 14 14 14 14 14 14  
15 15 15 15 15 15 15 15 15 15  
16 16 16 16 16 16 16 16 16 16

11. B, B, B, B, B, B, B  
 12. B, B, B, B, B, B, B, B, B, B  
 13. B, B  
 14. B, B

1970-71  
1971-72  
1972-73  
1973-74  
1974-75

19 ESD 0394 Q,L,E,I,E,O,D,D,I,I,I,I,  
J,O,D,E,E,S,G,O,O  
20 ESD 0395 Q,L,E,I,E,O,D,D,I,I,I,I,  
J,O,D,E,E,S,G,O,O  
21 ESD 0396 Q,L,E,I,E,O,D,D,I,I,I,I,  
J,O,D,E,E,S,G,O,O  
22 ESD 0397 Q,L,E,I,E,O,D,D,I,I,I,I,  
J,O,D,E,E,S,G,O,O  
23 ESD 0398 Q,L,E,I,E,O,D,D,I,I,I,I,  
J,O,D,E,E,S,G,O,O

## LISTINGS

## LISTINGS

## LISTINGS

## LISTINGS

# Binders

**Organise and protect your disk with Commodore Disk User disk binders and data disks.**

Why not keep your *Commodore Disk User* program collection alongside your magazines in a stylish *Disk User* disk binder? The binder comes complete with 10 disk sleeves to organize and protect your programs disks. Why not buy a disk binder to house all of your disk data? We can even supply *Commodore Disk User* data disks. The *Commodore Disk User* logo immediately identifies your disks and there's room to title them and document the disk contents.

#### **REFERENCES**

**Commissioner-Dkt User Banker (4-7), including  
all others - Order until 12/2003-03**

Comments: Order code ADP01

10 sheets, #7 Mylar®. Order code 800102  
10 sheets for insertion in binder, #134. Order code  
800103

20 choices for inclusion in binder. \$2.75. Order code



PRODUCT NAME	ORDER CODE	QUANTITY	PRICE
Overseas postage add £1.00			

All orders should be sent to: YOUR COMMODORE READERS' SERVICES,  
ARGUS SPECIALIST PUBLICATIONS, 9 MALL ROAD, HEMEL HEMPSTEAD,  
HERTS, Herts, SG1 1RA.

# Extending Basic

*Declaring and using labels in Basic is not as difficult as it may seem*

*By Burghard-Henry Lehmann*

**O**ne advantage of writing machine code programs with an assembler is that you can declare labels for jumps and branches. An example of this is "Speedy Assembler", written by yours truly, and still available from Reader Service for everybody who wants to get into serious machine code programming!

Instead of having to give jumps and subroutine calls as absolute addresses, and having to calculate relative addresses, you simply declare a label on the left-hand side at the start of the routine to be jumped or branched to, and the assembler does the rest. It stores the label in what is called "the symbol table", with the address of the location at that point next to it.

Later on, when it finds that label next to a jump or branch instruction, it fetches the address from the symbol table and, in case of a jump or subroutine call, assembles it as the location to be jumped or to be called to; in the case of a branch instruction, calculates the length of the branch. The programmer doesn't have to trouble himself with any of this. Nor does he have to go to the trouble of calculating branches. Instead he just attaches a name to the routine or subroutine in question.

And what's more, labels like this add tremendously to the readability of the program, because you can give each routine and subroutine a name that suits you. This helps quite a bit when a bug has developed in the

program (and doesn't it always?), and the programmer has to spend ages finding it!

## Labels in Basic

Because of all this, I felt for a long time that it would be nice to be able to use labels like this in Basic programs too. No more remembering of line numbers, and, most of all, no more remembering of GOTOs and GOSUBs whenever one changes the program.

It is actually surprisingly easy to introduce such a facility to the rather poverty-stricken Commodore 64 Basic. In the last article of this series we've developed a routine which allows us to give GOTO and GOSUB with variable names. So we've already got the basic facility to assemble jumps and subroutine calls, but we still have to declare the value of the label at the beginning of the program with a line

like "Subroutine = 1000". To do this job for us, the computer has to build a symbol table of sorts, and this has to be done before the program is actually run. This is because, during execution, when a GOTO or GOSUB is encountered, the computer has to know where to jump to.

You may know that most assemblers are called "Two Pass Assemblers". This is because in order to deal with jumps and branches, the assembler has to do its job in two goes. First it goes through the whole of the source and builds the symbol table, and then it has a second go in which it is able to assemble the source in earnest.

To use labels to the full in Basic, we have to do a similar thing. Before the program is run or earnest, the computer has to sift through the whole of the source and collect all the line numbers to which it has to GOTO and GOSUB later on. This means that it

Figure 1

	BasicRan	PHP	Save status register
A\$T1	LDA #	LDA #	This is direct ram mode
A\$T2	JSR LPP#	JSR LPP#	Set Kernel Flags to direct ram mode
A\$T3	PLP	PLP	Retrieve status register
A\$T4	BNR Line#	BNR Line#	If Z=0 then RUN plus line number
A\$T5	JMP	JMP	Do CLR and start program
ASTD	Line#	JSR SAMP	Do CLR
A\$B0		JMP SAMP	Jump to GOTO and start program

takes a bit of time until running proper starts. But I don't think that this is a major handicap, and it certainly is worth it!

### Running a Basic Program

So we have to intercept the RUN command. To do this, let's look first of all a bit closer at what it does under normal circumstances.

After you've typed in your Basic program and then given the RUN command to execute your program, the Basic interpreter jumps to the routine at \$A871. Figure 1 gives you a disassembly of that routine.

First the flag register is tested on the stack. Then a call to a Kernel routine is made. This routine puts the computer into the direct run mode by loading the system variable \$8D with zero and setting ST, which is the status variable.

Then the flag register is pushed from the stack again. If the zero flag equals one, that is, if the last value in the accumulator has been zero, then there were no parameters with the RUN command. In this case the routine continues, otherwise it branches forward, because a line number has been given with RUN.

If no line number is given with RUN the routine jumps to the CLR routine and doesn't return. If a line number has been given with RUN, the routine calls on the CLR routine and then jumps to the GOTO routine, because in the end, RUN 100 is similar to GOTO 100. The only difference is that RUN clears all the Basic variables, while GOTO leaves them untouched.

The CLR routine clears all the Basic variables and gives Basic a fresh start, so that variables and arrays can be built up anew. This is mainly done by setting the string storage pointer and the array storage pointer to the end of the Basic routine, because, as you might know, Basic stores all the variables and arrays declared in a program directly after the end of the routine.

### Modified RUN

Our modified RUN routine starts at line 1520 (Listing 1). First we deal with the RUN command more or less in the same way as the ROM routine. That is, we set the Kernel flags to the direct run mode (line 1520-1530), and then we clear the Basic variables (line 1540).

I haven't bothered about RUN line number, so this won't work with the routine as it is. If you give a line number with RUN, it will just be ignored. But nevertheless, if you do want this little-used facility, it shouldn't be impossible for you to add it with the help of the explanation of the normal RUN routine, which I've given above.

Next, we at lines page 234/235 to the start of the Basic routine (line 1589-1590). Zero page locations 231 to 235 are never used by the Commodore operating system, so these locations are absolutely safe to be used for your own purposes. There are many other zero page locations which are used by the Kernel or the Basic interpreter, but which are usually quite safe to use. For example, if you don't do any floating point arithmetic, you may use locations 561 to 565 without any trouble. But the point is always to think before you use a zero page variable, otherwise the system might do very funny things indeed!

After that we go into the main loop (SEARCHLP), which looks at each Basic line to see if it contains a label (line 1596-1610). To understand SEARCHLP, here is a short explanation of how a Basic line is stored in memory:

First of all the line number is given in the usual low byte/high byte fashion. Next there are two bytes which contain the so-called link pointers. Each link pointer points to the beginning of the next line. This makes it very easy to search through a Basic routine, because you just have to jump from one link pointer to the other, each time looking at the line number preceding it, and in next to no time you've found the line you're looking for.

Each Basic line is finished with a zero, which is the standard terminator used by Commodore (strings too are always terminated with zero!). At the end of the routine there will be two zeros in the locations where otherwise the next line number would be. So this is how the computer will know when it has reached the end of the routine.

If you look at SEARCHLP in our program, you'll find the routine testing for those two zeros right at the beginning of the loop (line 1616-1620).

### Declaring a Label

Naturally we have to tell the computer where it has found a label. For that

we have to make a label stand out in some way.

To do this I have chosen the following way of declaring a label: a label has to be at the beginning of a line, after the line number, and it has to be preceded by a full stop. Of course, you are free to experiment with methods which might suit you better, because this is the whole purpose of this series of articles, to enable you to develop extended Basic routines which fit your particular needs!

Anyhow, in the routine given, the computer looks for a full stop and that tells him that it has found a label (line 1666-1720). So it jumps to the routine which I called LABFOUND. First of all the current location in the routine is saved in STA/TH, so that afterwards the computer can continue searching the text for more labels (line 1666-1690).

Then it goes forward two bytes to point at the label itself (line 2080-2084). Remember, all we are doing here is a simple LET operation, like "LET LABEL = 1000" (or "LABEL = 1000") if you omit the LET. So the next ROM routine (\$B040) we call validates our label, that is, checks out if it is a permitted variable name (as you know, a valid Basic variable name has to start with a letter).

If the name is valid, the first two characters of it are stored in the variables area, which starts immediately after the end of the Basic routine. On return from \$B040, the low byte of the variable location is in the accumulator, and the high byte is in the Y-Register. We then have in zero page \$40/44, which is the system variable pointer (line 2110-2120).

Now we get the line number and store it in zero page \$42/43, which is the floating point accumulator #1 (line 2140-2145). The next routine (\$B359) which we call converts the line number into a proper floating point number (line 2220). We have to do it in this rather roundabout way because variables of this type have to be stored as floating point numbers in order to be recognized later on!

Finally we call \$B040 (line 2260), which moves the value contained in register #1 in the variable area load. Now our label has been stored like any other Basic variable in the variable store, including the number of the line on which it appeared. We can return to our main search loop and look for the next label.

The fact that each of the labels we

declare is stored like any other Basic variable name, of course, that only the first two characters of the label will be recognised. The rest of the label is ignored, which limits the use of labels

rather, because it doesn't give much scope for declaring meaningful names which are recognised.

Therefore, in the next article I'd like to develop a routine which builds

and recognises a symbol table away from the basic variable area. This will enable us to declare labels with six or more characters which are fully recognised.

### Listing 1

10	ORG 48000	
20	CNT	
30	JOB CHARGEMT	
40	JOB BIASFA	
50	JOB BIASFT	
60	JOB BIASBT	
70	:	
80	;	
90	THEN EXTENDED BASIC ON	
100	JMP CHANGING VECTOR AT 80000	
110	;	
120	EXTROUTN LDA P-FREIGHTANT	
130	STA C-DIRECTCT	
140	LDA P-FREIGHTANT	
150	STA C-DIRECTCT	
160	:	
170	RTS	
180	:	
190	;	
200	;	
210	THEN EXTENDED BASIC OFF	
220	JMP CHANGING VECTOR AT 80000	
230	BACK TO NORMAL (BASIC)	
240	;	
250	EXTROUTN LDA P-BIASFA	
260	STA C-DIRECTCT	
270	LDA P-BIASFA	
280	STA C-DIRECTCT	
290	:	
300	RTS	
310	:	
320	;	
330	;	
340	*** MAIN PROGRAM ENTRY ***	
350	;	
360	LOOK FOR EXTENDED BASIC COMMANDS	
370	;	
380	FREIGHTANT JMB CHARGEMT	
390	JMB BIASFT	
400	JMB BIASBT	
410	:	
420	;	
430	BIGSMITH CMP 'O'	
440	BIG HLT	
450	JMP CDT BT	
460	NEXT CMP #8000 :	GOTO
470	BIG GOTO BT	
480	CMP #8000 :	GOTO
490	BIG HLT	
500	JMP HLT BT	
510	NEXT1 CMP #8000 :	GOTO
520	BIG HLT BT	
530	CMP ' '	SPECIAL
540	BIG SPCL BT	
550	:	
560	CMP 'C'	
570	BIG NORMAL	
580	JMB CHARGEMT	
590	CMP 'D'	
600	BIG NORMAL	
610	JMB CHARGEMT	
620	CMP 'L'	
630	END NORMAL	
640	JMB CHARGEMT	
650	CMP #8000 :	GOTO
660	BIG COLOR BT	
670	:	
680	;	
690	;	
700	;	
710	;	
720	;	
730	;	
740	;	
750	EXECUTE "COLOR" COMMAND	
760	;	
770	;	
780	;	
790	;	
800	;	
810	;	
820	;	
830	;	
840	;	
850	;	
860	;	
870	;	
880	;	
890	;	
900	;	
910	;	
920	;	
930	;	
940	;	
950	;	
960	;	
970	;	
980	;	
990	;	
1000	;	
1010	;	
1020	;	
1030	;	
1040	;	
1050	;	
1060	JMP TO EXIT OF NON-Routine	
1070	:	
1080	;	
1090	;	
1100	;	
1110	;	
1120	;	
1130	;	
1140	;	
1150	;	
1160	;	
1170	;	
1180	;	
1190	;	
1200	;	
1210	;	
1220	;	
1230	;	
1240	;	
1250	;	
1260	;	
1270	;	
1280	;	
1290	;	
1300	;	
1310	;	
1320	;	
1330	;	
1340	;	
1350	;	
1360	;	
1370	;	
1380	;	
1390	;	
1400	;	
1410	;	
1420	;	
1430	;	
1440	;	
1450	;	
1460	;	
1470	;	
1480	;	
1490	;	
1500	;	
1510	;	
1520	;	
1530	;	
1540	;	
1550	;	
1560	;	
1570	;	
1580	;	
1590	;	
1600	;	
1610	;	
1620	;	
1630	;	
1640	;	
1650	;	
1660	;	
1670	;	
1680	;	
1690	;	
1700	;	
1710	;	
1720	;	
1730	;	
1740	;	
1750	;	
1760	;	
1770	;	
1780	;	
1790	;	
1800	;	
1810	;	
1820	;	
1830	;	
1840	;	
1850	;	
1860	;	
1870	;	
1880	;	
1890	;	
1900	;	
1910	;	
1920	;	
1930	;	
1940	;	
1950	;	
1960	;	
1970	;	
1980	;	
1990	;	
2000	;	
2010	;	
2020	;	
2030	;	
2040	;	
2050	;	
2060	;	
2070	;	
2080	;	
2090	;	
2100	;	
2110	;	
2120	;	
2130	;	
2140	;	
2150	;	
2160	;	
2170	;	
2180	;	
2190	;	
2200	;	
2210	;	
2220	;	
2230	;	
2240	;	
2250	;	
2260	;	
2270	;	
2280	;	
2290	;	
2300	;	
2310	;	
2320	;	
2330	;	
2340	;	
2350	;	
2360	;	
2370	;	
2380	;	
2390	;	
2400	;	
2410	;	
2420	;	
2430	;	
2440	;	
2450	;	
2460	;	
2470	;	
2480	;	
2490	;	
2500	;	
2510	;	
2520	;	
2530	;	
2540	;	
2550	;	
2560	;	
2570	;	
2580	;	
2590	;	
2600	;	
2610	;	
2620	;	
2630	;	
2640	;	
2650	;	
2660	;	
2670	;	
2680	;	
2690	;	
2700	;	
2710	;	
2720	;	
2730	;	
2740	;	
2750	;	
2760	;	
2770	;	
2780	;	
2790	;	
2800	;	
2810	;	
2820	;	
2830	;	
2840	;	
2850	;	
2860	;	
2870	;	
2880	;	
2890	;	
2900	;	
2910	;	
2920	;	
2930	;	
2940	;	
2950	;	
2960	;	
2970	;	
2980	;	
2990	;	
3000	;	
3010	;	
3020	;	
3030	;	
3040	;	
3050	;	
3060	;	
3070	;	
3080	;	
3090	;	
3100	;	
3110	;	
3120	;	
3130	;	
3140	;	
3150	;	
3160	;	
3170	;	
3180	;	
3190	;	
3200	;	
3210	;	
3220	;	
3230	;	
3240	;	
3250	;	
3260	;	
3270	;	
3280	;	
3290	;	
3300	;	
3310	;	
3320	;	
3330	;	
3340	;	
3350	;	
3360	;	
3370	;	
3380	;	
3390	;	
3400	;	
3410	;	
3420	;	
3430	;	
3440	;	
3450	;	
3460	;	
3470	;	
3480	;	
3490	;	
3500	;	
3510	;	
3520	;	
3530	;	
3540	;	
3550	;	
3560	;	
3570	;	
3580	;	
3590	;	
3600	;	
3610	;	
3620	;	
3630	;	
3640	;	
3650	;	
3660	;	
3670	;	
3680	;	
3690	;	
3700	;	
3710	;	
3720	;	
3730	;	
3740	;	
3750	;	
3760	;	
3770	;	
3780	;	
3790	;	
3800	;	
3810	;	
3820	;	
3830	;	
3840	;	
3850	;	
3860	;	
3870	;	
3880	;	
3890	;	
3900	;	
3910	;	
3920	;	
3930	;	
3940	;	
3950	;	
3960	;	
3970	;	
3980	;	
3990	;	
4000	;	
4010	;	
4020	;	
4030	;	
4040	;	
4050	;	
4060	;	
4070	;	
4080	;	
4090	;	
4100	;	
4110	;	
4120	;	
4130	;	
4140	;	
4150	;	
4160	;	
4170	;	
4180	;	
4190	;	
4200	;	
4210	;	
4220	;	
4230	;	
4240	;	
4250	;	
4260	;	
4270	;	
4280	;	
4290	;	
4300	;	
4310	;	
4320	;	
4330	;	
4340	;	
4350	;	
4360	;	
4370	;	
4380	;	
4390	;	
4400	;	
4410	;	
4420	;	
4430	;	
4440	;	
4450	;	
4460	;	
4470	;	
4480	;	
4490	;	
4500	;	
4510	;	
4520	;	
4530	;	
4540	;	
4550	;	
4560	;	
4570	;	
4580	;	
4590	;	
4600	;	
4610	;	
4620	;	
4630	;	
4640	;	
4650	;	
4660	;	
4670	;	
4680	;	
4690	;	
4700	;	
4710	;	
4720	;	
4730	;	
4740	;	
4750	;	
4760	;	
4770	;	
4780	;	
4790	;	
4800	;	
4810	;	
4820	;	
4830	;	
4840	;	
4850	;	
4860	;	
4870	;	
4880	;	
4890	;	
4900	;	
4910	;	
4920	;	
4930	;	
4940	;	
4950	;	
4960	;	
4970	;	
4980	;	
4990	;	
5000	;	
5010	;	
5020	;	
5030	;	
5040	;	
5050	;	
5060	;	
5070	;	
5080	;	
5090	;	
5100	;	
5110	;	
5120	;	
5130	;	
5140	;	
5150	;	
5160	;	
5170	;	
5180	;	
5190	;	
5200	;	
5210	;	
5220	;	
5230	;	
5240	;	
5250	;	
5260	;	
5270	;	
5280	;	
5290	;	
5300	;	
5310	;	
5320	;	
5330	;	
5340	;	
5350	;	
5360	;	
5370	;	
5380	;	
5390	;	
5400	;	
5410	;	

## PROGRAMMING

### **Lysine 23**

```

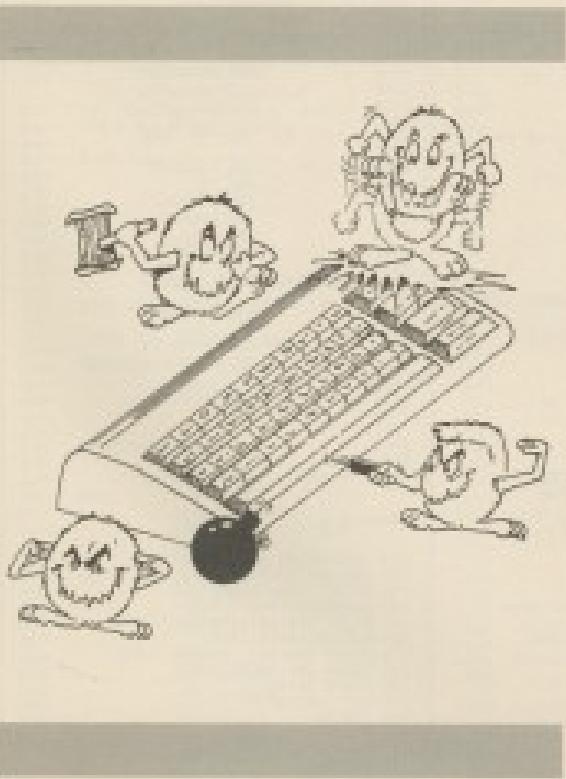
110 REMAINING
120 NEW-LINE(CHR$(14)+CHR$13)
130 PRINT "PAGE", PAGE
140 PAPER=INT(PAGE/1000)
150 PRINT TAB(15); "PAGE"; PAPER;
160 END-(INT(PAGE/1000))
170 FRONT PAGE="NUMBER"; END
180 COLOR INK; PAPER,END
190 COLOR RELAY
200 COLOR RELAYEND
210 COLOR
220 FOR NOL TO 600
230 NEXT
240 RETURN

```

# Tech Troubles

*A selection of the problems solved and readers' hints  
from this month's mailbag*

*By Andy Andras*



**W**hy do computers use integers as well as normal decimal values, and where do floating point numbers come into it?

David Kennedy, Eastgate

Dear David,

Another term for an integer is a 'whole number' or, as you point out, a number without any decimal point. Other numbers are referred to as 'floating point' numbers because of the way in which the mathematics is performed in the computer's memory. Floating point calculations are different to those in the space allowed here but, luckily, the fact that they do work is all that you really need to know for now.

The reason that computers use integers is because it helps to speed up arithmetic, and is more accurate than floating point when a certain amount of rounding up and down of values occurs. The speed benefit is gained because there is no need to calculate the position of the decimal place, or to check if the value should be expressed in exponential form, when an extremely large or small number is expressed by a value, and the power to which it has to be raised to reveal the actual number.

A popular myth is that integers use less memory than floating point values, but this is not entirely true. Integers are four bytes - two for the variable name, and two for the actual number. Floating point numbers similarly use two bytes for the variable name, but five more bytes are needed for the value. So, on the face of it,

integers do need less space, but the integer value is followed by three unused bytes, which means that both types of variable physically occupy several bytes of memory.

I thought the *Tutor Commodore 64 Series User Guide* 1988, and found the *dit tips for the d6* quite informative. One of the tips is how to simulate a PRINT AT command without using rows of cursor down and lefts. I thought that you may be interested in a technique that I've been using for the past few years:

```
10: POKE 144,8:PRINT "LINE 1"
20: POKE 144,10
30: PRINT "LINE 2"
```

*Location 284 is the vertical position of the cursor on the screen (ruler 1 to 24). The PRINT command in line 10 is used to update this new row in the Commodore's memory, and a cursor up or replace the cursor onto the required line. Next the cursor is placed horizontally by poking a value in location 217 and the message can then be printed.*

*Two final words on this exciting feature is printing on the top and bottom lines. For using the top line, printing a cursor home with a semicolon will set the row position and then the column value can be poked into 217. If the last line is used the cursor will scroll up until you follow the message printing command with a semicolon.*

*I hope this is of use to someone, somewhere.*

Carol Seddon, Wigan

Dear Carol,  
Thanks very much for that tip. I found it very informative, and if anyone else has a technical tip I'd be very pleased to hear about it.

In a listing I came across the following syntax which I don't understand:

```
100: RECODE CHRS(144)+POKE 144,
```

Could you explain what it achieves?  
Clive Foster, Midhurst

in D6-T6 until the correct point is reached?

Harry Ivey, Hampton

Dear Clive,

This is an unusual but useful application of a comparison command. It is the sort of routine which would convert a decimal value to its hexadecimal equivalent.

To understand how it works, I'll have to explain the principle behind the *greater than* and *less than* commands. When a comparison is made, the result is a zero if the condition is false and a minus one if the condition is true.

In the example that you've supplied, the value given by the comparison would be zero if X had a value between nine and nine plus minus one if the value exceeds nine. What the equation is doing is taking the value in A and adding 48 to give a value whose CHR\$ equivalent is the number character which corresponds to X. This works well up to a value of nine, but values 10 to 13 would produce punctuation marks and symbols because there is a gap of seven characters between the CHR\$ value for nine and the CHR\$ value for the letter A. The program has to be able to add seven to values over 10, but not to values below that.

The solution is to compare the value with nine, and then multiply this by seven. This would mean that the low values would subtract zero from the CHR\$ value but high values would subtract -7 (-1 multiplied by 7). Subtracting a negative value follows the mathematical law that states that two minuses gives a plus, so the program actually adds seven to the CHR\$ value.

To test this try setting up a loop for A having values of zero to 12. Place the equation inside the loop and a statement to print each HEXA value, and the result will be a list of hex equivalents from 0 to F.

I have written a program which uses several groups of D6-T6 statements but the only problem with it is that RESTORE can't be used to point to a particular line. Is there any way around this, because at the moment I have to RESTORE and then read

Dear Harry,

There are two ways to tackle your problem. The first is to read the data into several arrays and manipulate these instead of using data READs.

The second method is to create a special form of the RESTORE command. This is done by reading in the data until the line where the program is to be restored. The start memory location can be stated by poking locations 65 and 66 and storing this in a table as a save part of memory.

Each suitable point is similarly stored until the table is complete. When the data reading has to be restored to any of these points, they can be peaked from their storage point and peaked back to 65 and 66.

Can you tell me where I can find a good book containing a breakdown of the C128 BASIC ROM? I have a book that such a book may not exist. Because I have a book which gives a breakdown of the Kernel, but which states that the basic assembly would fill a very large volume.

Richard Trey, York

Dear Richard,

You'll be pleased to hear that such a book does actually exist, though you are right in assuming that it is a weighty tome. C128 BASIC 7.0, formerly published by Abacus, runs to over 600 pages. Of this, 450 pages contains an annotated ROM disassembly, and the rest is packed with useful information about the workings of the system, alongside some useful programming hints and routines. The price is £16.99, and it is available from Precision Software, 6 Park Terrace, Woodstock Park, Surrey KT4 1JZ, but please include £1.89 for post and packing.

If you have a problem let us know and Andy will try to help. Write to Tech Troubles, Your Commodore, Argus House, Boundary Way, Belmont Hengistwood, Herts HP1 1ST.

# THE EPSON SQ-2500

The first major task is getting the SQ-2500 out of the box - ideally, it's a job for two people. The SQ-2500 might be expensive, but it has a wide carriage and you certainly get a lot of printer for your money. Styling is recognisably Epson, but the greatest impression from the first is that this printer costs quality. Construction is superb. I didn't actually try it, but I'm sure I could jump on the SQ-2500 without damaging it (and I weigh over 10 stones).

## Setting it up

This is an ink-jet printer so there's no ribbon, just an ink cartridge, paper feed knob, the main lead and paper guide. Open a cover on the right side towards the rear of the printer, push in the ink cartridge, close the cover and that's done. On the left side is a similar cover, but unless you want to use either of the possible two extra font cartridges, you needn't open it.

The interface and power lead connect at the rear, as usual for Epsons, and the power switch is on the right side. The paper guides easily slot into position, and when an new ink cartridge is fitted, the SQ needs printing jobs in the long cleaning cycle mentioned later, but it just means switch on and press a button. After about half a minute it's ready to go.

Both parallel and a serial interface are standard, with provision for a third one, and any of the seven Epson LQ-Interface (listed in the LQ-580 report) can be used too. The standard buffer is 32K. The manual is 178 pages, in contrast to the LQ's 256, but contains all the necessary ingredients, including both types of command summary and a quick reference card.

If you can afford an SQ, buy one! If you do, don't waste time looking for accessories - there aren't any. Four buttons (and one of these is an all font control **EVERYTHING!**) This explains the short manual. There's very little on trouble-shooting,

*To conclude last month's feature on the world of 24 pin dot matrix printers,  
Robin Burton admires the highly impressive  
Epson SQ-2500*

The only way I can envisage anyone having trouble with the SQ is if they don't read, in which case the manual wouldn't help much.

I'm not being flippanç, it's all set-up, and manual selection is by question and answer, but with a difference. There's a liquid crystal display at the front of the printer next to the selection buttons, and the SQ asks the questions. You answer by simply pressing a button, and the settings are stored. This is a clever machine.

So far though this is, there's more! The SQ also has four macros. Each of these is a complete, permanently stored definition containing everything you could want the printer to know about a job. Any one of the four can be loaded automatically at power-up; you choose which one you want in the configuration details, which remain permanently set unless you alter them.

You can manually load any macro by a couple of button presses, and can also amend them manually, by software or hard, either temporarily, or permanently. Simply re-save a macro at any time by pressing a button if you want the changes to be permanent. If not, they're forgotten, either when you switch off or when you load a different macro. You might also want to print out the settings, in which case - press a button.

This is all so comprehensive yet simple that describing it is difficult. I've therefore included a print of any configuration settings separately, so you can see for yourself. Everything is stored - left and right margins, font, style, pitch, page size, etc, etc.

As standard, the SQ-2500 is a car sheet machine, but an optional tractor unit can be added for continuous paper. You may have gathered by now that the SQ-2500 is intended for high quality, high speed, high volume output. The fact that the tractor is unmounted and therefore without paper parking is, in context, irrelevant.

Don't misunderstand - switching between continuous and sheet fed is very enough, but if you tend to do it very often you don't need an SQ-2500. It wasn't built for the average home user's mixed low-volume needs, and used that way would be like doing the grocery shopping in a Formula One car, theoretically possible but...

If you sheet is the major use a double-bin automatic sheet feeder can also be attached.

## Specification

The SQ-2500 is of course above all an Epson, so the compatibility comments for the LQ-580 all apply to the SQ, including the 12 national character sets.

Six fonts are standard for the SQ-2500, such as used with most. Additional effects are limited to double width, double height or both. All fonts are available in the usual 10, 12 and 15 cpi, and all can be condensed except 15 pica.

There's also provision for two font modules, but so far as I could see all the Epson LQ fonts are standard in the SQ except OCR-B. Minimum vertical spacing is in 180ths of an inch, and horizontal is in 360ths.

## Using the Epson

When the SQ powers-up, it automatically goes through a self-cleaning cycle. The printer informs you (via the LCD) where the ink cartridge is going low, and when it's exhausted. According to the manual, a cartridge lasts for 5,000,000 characters in LQ, and 6,000,000 in draft. This seems fair enough - I've gone through well over 1,000 sheets, largely in LQ, (roughly 6,000,000 characters) and not a word so far.

Using semi-automatic sheet feed is simple and quick, and it has been adequately reliable with all weights of paper. Just drop a sheet into the guide, press 'form feed' and the sheet is lined up perfectly (I even tried it with a 9 x 4 inch envelope - notoriously difficult to keep straight - with no trouble at all).

If the receiver is fitted, it simply attaches on top of the SQ - there are no covers to change. Unfably, along with the tray, Epsons provide a matching base on which the printers sit. Continuous paper is kept inside this, out of the way. A paper rest is also included to keep the paper clear of the leads.

Manual line feed, form feed, self-test and hex dump are controlled by the buttons, and the self test includes the current configuration and status settings. As mentioned before, just the configuration and the four macros can also be listed on demand (or alternatively selected directly via the LCD). Without altering any of the current settings you can also switch between draft or LQ at any time by pressing either the line-feed or the form-feed buttons.

Print quality is frankly so superior to that of an impact dot-matrix printer that a comparison is pointless. Because there are no wires passing through a fabric ribbon, the individual dots are much smaller and more precise. I've

even asked when I bought a laser printer by people who didn't know I had the SQ. It might not be quite that good, but it's obviously near enough, and you'd have to put the two side by side to notice a difference. Also unlike any impact printer, the quality never varies. There's no deterioration or fading of print quality, because there's no ribbon to wear out. Characters are always perfect, and uniformly nicely black.

The SQ is the fastest matrix printer I've ever seen. The figures don't adequately tell the story - you have to see it to appreciate just how quick it is. In fact its speed was at one point an inconvenience. With such dense character images, double strike is almost unnecessary. Of course I tried it, but output was quick enough to allow the paper to re-fold in the collection basket before the ink had fully dried. I have now lowered the basket.

The final difference about the SQ is noise, specifically the absence of it - only ink touches the paper. You don't realise just how much you are mentally conditioned to brace yourself for the onslaught of noise from an impact printer, at least not until you start printing and it doesn't happen. It's rather uncanny at first, and after years of impact printers, it took me a week to get used to this. Needless to say the SQ easily passes the telephone test, in fact if you were surrounded by them it would be no problem.

Quite aside from its obvious desirability, this aspect of the SQ's performance (along with the others) offers very serious benefits for offices where several printers are in use. I would think the amount of noise from the SQ (just the carriage moving back and forth) would be acceptable even in places like libraries and hospitals too.

The SQ-2000 has operated perfectly throughout.

## Conclusion

It hardly needs saying that the SQ-2000 is Epson's top-of-the-range conventional dot matrix printer, and it has been for a few years now. It is built like the Forte Bridge, and has by far the easiest manual control of all operational features of any printer I know. Macros are excellent, and allow an automated switch between four completely different setting definitions in literally a couple of seconds.

Of course the recent arrival of low-cost laser printers must have eroded SQ sales, but there are still plenty of jobs that laser printers don't do very well (or at all: like printing A3 pages sideways), and their running costs are also higher than the SQ's. One laser printer limitation however suffered by the SQ-2000, because it's not an impact device, multi-part sets are impossible, though with its speed you'd probably just print extra copies as needed.

In its target market, I doubt that the SQ-2000 has any competitor. Quality is beyond question in construction, operation and output. It need be the fastest, quietest way of getting high quality print onto paper, with all the flexibility of a conventional printer and a virtual absence of operator skill or training.

The recommended retail price is £1,349. Options include a tractor unit at £20 and a double-bin automatic sheet feeder at £179 (£1,291.25, £160.50 and £102.50 including VAT.) Ink cartridges are about £24, and have a shelf life of two years.

Checking current advertisements I found that the SQ-2000 can be purchased for around £975 plus VAT, with proportional reductions in the optional fittings.

## TABULATION FOR YOUR COMMODORE/MAY/PRINTER<sup>2</sup>:

	Dimensions WxDxH	Time/Speed to print 5000 chars		Weight
		Draft	Letter	
Star LC24-98	175x175x27"	56 sec/59 cps	128sec/56 cps	14.5 Lbs
Epson LX-800	16.3x12.7x8"	53 sec/54 cps	129sec/50 cps	15.5 Lbs
Calico HQP-105	16.1x14.7x7.7"	45 sec/101 cps	180 sec/50 cps	16.5 Lbs
Epson SQ-2000	23.3x15x8	32 sec/156 cps	56 sec/89 cps	25 Lbs

# Sketchpad 128



*Gordon Davis examines a rare specimen - an 80-column graphics package for C128 owners*

**S**ketchpad 128 is a rare animal. Almost all graphics packages for the C128D operate in 80-column high-resolution mode, but this one is different - it offers full graphics capability in 40-column mode - a resolution of 640 pixels horizontally.

The package has been made possible by the release of Winkworth's Basic 8, a language package which we reviewed last year. Basic 8 is designed as a language which offers a graphics-oriented environment to the system designer. It's no surprise, therefore, that the same features of Sketchpad are exactly those which Basic 8 supports most readily.

A Commodore 128 mouse is a necessity for Sketchpad. This, I think,

is unfortunate, since it isn't the sort of item that Commodore owners are likely to just have hanging around. Although, as the authors point out, a mouse is by far the best device for doing graphics, it would have been little trouble to allow a joystick to be interfaced with the program, just to get people started.

### Little messing

Fortunately, we managed to find the off-the-shelf 128 mouse, so the play could begin. There's very little messing around following loading. Sketchpad goes straight into the drawing screen following the obligatory BASIC 8 loading screen.

Free Spirit, which, as you might guess, is a California software house, has obviously done its Apple Macintosh homework. The main screen looks very Mac-like. All menu selections are made from the drawing screen by clicking the left mouse button. The right button is reasonably consistent in returning the user to the main menu.

The main elements on the main menu are exactly those which have been made familiar by that unusual program, MacPaint. Naturally, you can draw freehand on the screen with a variety of different pens. The indispensable eraser is also there. The annoying thing about this is that the artwork area cannot be saved, which is something that you can see even with C64 packages.

Sketchpad supports a variety font file. This is a little messy on the way it operates, since it seems to leave many files scattered along the way. Not never fear, the filing algorithm gets back and files them all back in too.

A multi-line option for writing is also available. Different fonts can be loaded from disk, and can be typed anywhere on the screen in one of ten different sizes. A useful touch is that Return will align the next cursor exactly one line below the one you've already used.

This is not a multi-colour graphics program, quite rightly, because unless you are designing loading screens or demos, anything other than monochrome screens is functionally useless.

for all owners of black-and-white printers. Even so, you can alter the foreground/background to give you more creative flexibility.

Although creating the scenes is fairly deadly on Standard - there is no check on size, you can save your program into memory, which will put it into a space area of RAM for safety. This is also useful if you've inserted some of the changes you're making to a picture.

The graphics shapes option are the ones that show the main difference over most graphics packages. All the traditional favourites are there: bars, boxes, circles, and so on. Programming them is more or less as straightforward as the commands in any other graphics package. A unique extra, however, is the addition of the so-called Cylinder shapes.

### Solid graphics

Cylinder shapes are views of shaded solids of three types: cylinder, sphere and torus. They are created after the programme has designed a set of

algorithms for representing them easily.

Basic 8 later incorporated these calculations, and today they are at last in a graphics package. The shading on the Cylinder shapes can be varied between pixel dots or cross shading, and whatever you use them for, they are an interesting addition.

Besides these shapes, another useful feature is the inclusion of arcs. You can provide arcs of any pitch and angle. Rays can also be provided, subtended at any angle from a central point. Both of these makes the production of pictures an easy matter.

The final thing I must say is looking for was some form of Zoom option. It's very unsatisfactory using a graphics package, and some means of adjusting things on a pixel-by-pixel basis can be useful. This is, amazingly, not available from the main screen, but from a sub-menu called the Gadgets menu.

Besides a pixel zoom, Gadgets also supports a Cut/Paste option which allows you to move chunks of the picture around, or simply remove them. You can also scroll your picture - an interesting feature which the

package supports. This menu also supports a rotation of pictures which can be useful sometimes. Unfortunately, you have to drag the clip patterns in yourself - there is no automatic fill for these.

Fox Spirit deserves complimenting for the number of features which have been included in Standard - it hasn't covered everything here. The main flaw in the program is that it is a little slow, due to being written entirely in Basic 8.

I'm also not sure about some of the design priorities - some of the essentials are very hard to get to. The program documentation is brief and unhelpful. These days, this is something I expect software houses to get right. On the other hand, C128 add-on drivers' sharing packages are fairly hard to come by, so it seems unfair to criticise much.

### To conclude:

Fox Standard 128. Supplier: FOX SPIRIT SOFTWARE, 21 High St., Wootton, Wiltshire, SN14 7BG. Tel: 01249 534155. Price: £49.95.

**DON'T GET IN TANGLE...**

**take out a subscription**

Make sure of your regular copy of Micro Music, with all the news, signs and reviews from the computer music scene, delivered to your door at no extra cost in the UK. Just fill in the form, enclose your remittance and you're in touch.

Address:	UK:	£18.50
Address:	Europe:	£14.00
Address:	Mobile Post:	£14.15
Address:	Far East:	£13.50
Address:	Rest of the World:	£14.00
Address:	USA:	\$25.00

Air Mail rates given  
on request. Overseas  
subscriptions include  
postage.



Micro Music  
Subscription  
Services  
Dept. MMS  
2000, 2002, 2004,  
2006, 2008, 2010,  
2012, 2014, 2016,  
2018, 2020, 2022,  
2024, 2026, 2028,  
2030, 2032, 2034,  
2036, 2038, 2040,  
2042, 2044, 2046,  
2048, 2050, 2052,  
2054, 2056, 2058,  
2060, 2062, 2064,  
2066, 2068, 2070,  
2072, 2074, 2076,  
2078, 2080, 2082,  
2084, 2086, 2088,  
2090, 2092, 2094,  
2096, 2098, 2100,  
2102, 2104, 2106,  
2108, 2110, 2112,  
2114, 2116, 2118,  
2120, 2122, 2124,  
2126, 2128, 2130,  
2132, 2134, 2136,  
2138, 2140, 2142,  
2144, 2146, 2148,  
2150, 2152, 2154,  
2156, 2158, 2160,  
2162, 2164, 2166,  
2168, 2170, 2172,  
2174, 2176, 2178,  
2180, 2182, 2184,  
2186, 2188, 2190,  
2192, 2194, 2196,  
2198, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
2268, 2270, 2272,  
2274, 2276, 2278,  
2280, 2282, 2284,  
2286, 2288, 2290,  
2292, 2294, 2296,  
2298, 2200, 2202,  
2204, 2206, 2208,  
2210, 2212, 2214,  
2216, 2218, 2220,  
2222, 2224, 2226,  
2228, 2230, 2232,  
2234, 2236, 2238,  
2240, 2242, 2244,  
2246, 2248, 2250,  
2252, 2254, 2256,  
2258, 2260, 2262,  
2264, 2266, 2268,  
2270, 2272, 2274,  
2276, 2278, 2280,  
2282, 2284, 2286,  
2288, 2290, 2292,  
2294, 2296, 2298,  
2200, 2202, 2204,  
2206, 2208, 2210,  
2212, 2214, 2216,  
2218, 2220, 2222,  
2224, 2226, 2228,  
2220, 2222, 2224,  
2226, 2228, 2230,  
2232, 2234, 2236,  
2238, 2240, 2242,  
2244, 2246, 2248,  
2250, 2252, 2254,  
2256, 2258, 2260,  
2262, 2264, 2266,  
226

# Routine Programming

*A bubble routine  
to sort lists into  
orderly sequences*

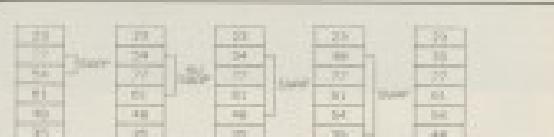
By Eric Doyle

Programs often include lists of numbers or strings which have to be displayed in an orderly way. This could be alphabetical or numerical. The bubble sort subroutine can be used, suitably modified, for either purpose.

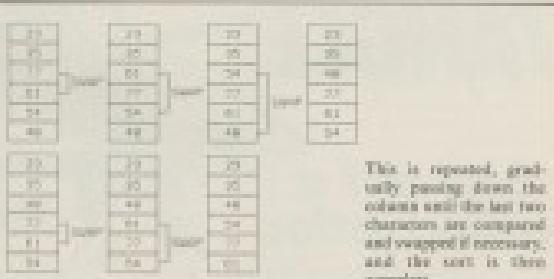
Bubble sorts work by comparing neighbouring list items and swapping them over if one exceeds the other. Take this list as an example:



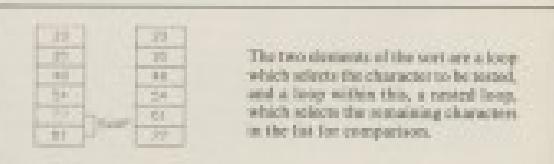
The first number is compared to the rest of the numbers in the list one at a time. If the number under comparison is smaller than the first number, a swap takes place. In this way the smallest value rises to the top.



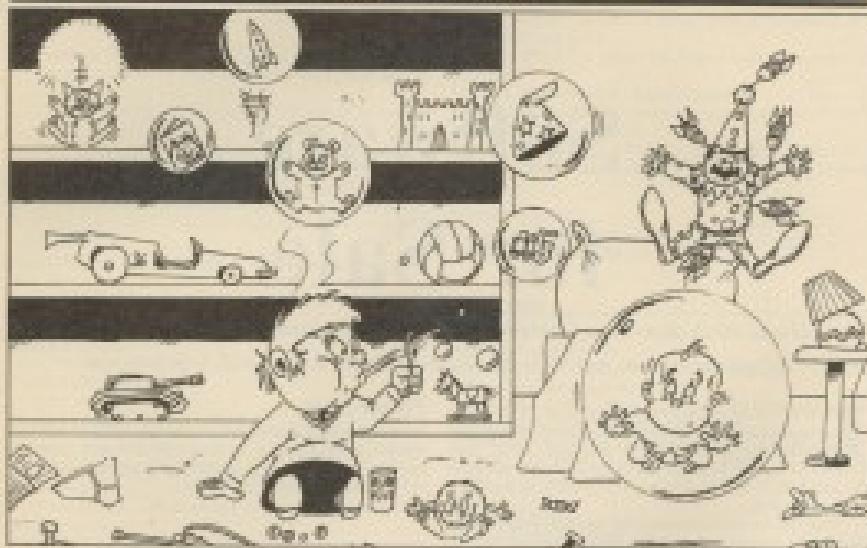
After this process is completed, the second number is compared to all of the following figures in a similar way.



This is repeated, gradually passing down the column until the last two characters are compared and swapped if necessary, and the sort is then complete.



The two elements of the sort are a loop which selects the character to be tested, and a loop within this, a nested loop, which selects the remaining characters in the list for comparison.



60200: FOR201-1002-1

60201: 200-201-1

60202: FOR203-20202

The value 2 is the number of items in the list. Line 60200 forms the main loop for the chapter to be tested. It only runs from the first to the index-punctuation character because there would be no point in comparing the final character with itself.

Line 60210 calculates the first character to be tested by the nested loop which is initiated in line 60200. Line 60220 tests the two values, and if they are already in the correct order no action is taken and the exchange routine is bypassed.

60220: IF201&gt;20200:THEN20200

If an exchange is necessary, this is done by using a temporary store for one of the values, so that values can be swapped with ease.

60230: 200-201-20211

60231: 20200-20200

60232: 20120-20120

200 takes the value held in Z(2) and 20200 can then be given the value held in 200. The stored value can then be moved from 200 to 20200, and 200 may then be discarded.

Now that everything has been done, the loops can be closed and

eventually control is handed back to the program with a RETURN command.

60240: NEXT201-21

60250: RETURN

The listings shown contain an example of numerical sorting using this technique and there is also an alphabetical sort routine which shows how this routine has to be modified for this use. Really all that has happened is that the Z(1) values are changed to Z(2).

## Parameters for Main Program

### Sending

Z Number of list items  
200 List item values

### Returns

200 The sorted list

### Other Variables Used

Z1 Main loop variable  
Z2 Remainder of list items  
Z3 Nested loop variable

#### PROGRAM NUMBER 20200

```
000: LD 0-1 200-201-100200,200001
001: 20 20 200001,100200-1
002: 20 20 200001,100200-1
003: 20 20 200001,100200-1
004: 20 20 200001,100200-1
005: 20 20 200001,100200-1
006: 20 20 200001,100200-1
007: 20 20 200001,100200-1
008: 20 20 200001,100200-1
009: 20 20 200001,100200-1
010: 20 20 200001,100200-1
011: 20 20 200001,100200-1
012: 20 20 200001,100200-1
013: 20 20 200001,100200-1
014: 20 20 200001,100200-1
015: 20 20 200001,100200-1
016: 20 20 200001,100200-1
017: 20 20 200001,100200-1
018: 20 20 200001,100200-1
019: 20 20 200001,100200-1
020: 20 20 200001,100200-1
021: 20 20 200001,100200-1
022: 20 20 200001,100200-1
023: 20 20 200001,100200-1
024: 20 20 200001,100200-1
025: 20 20 200001,100200-1
026: 20 20 200001,100200-1
027: 20 20 200001,100200-1
028: 20 20 200001,100200-1
029: 20 20 200001,100200-1
030: 20 20 200001,100200-1
031: 20 20 200001,100200-1
032: 20 20 200001,100200-1
033: 20 20 200001,100200-1
034: 20 20 200001,100200-1
035: 20 20 200001,100200-1
036: 20 20 200001,100200-1
037: 20 20 200001,100200-1
038: 20 20 200001,100200-1
039: 20 20 200001,100200-1
040: 20 20 200001,100200-1
041: 20 20 200001,100200-1
042: 20 20 200001,100200-1
043: 20 20 200001,100200-1
044: 20 20 200001,100200-1
045: 20 20 200001,100200-1
046: 20 20 200001,100200-1
047: 20 20 200001,100200-1
048: 20 20 200001,100200-1
049: 20 20 200001,100200-1
050: 20 20 200001,100200-1
051: 20 20 200001,100200-1
052: 20 20 200001,100200-1
053: 20 20 200001,100200-1
054: 20 20 200001,100200-1
055: 20 20 200001,100200-1
056: 20 20 200001,100200-1
057: 20 20 200001,100200-1
058: 20 20 200001,100200-1
059: 20 20 200001,100200-1
060: 20 20 200001,100200-1
061: 20 20 200001,100200-1
062: 20 20 200001,100200-1
063: 20 20 200001,100200-1
064: 20 20 200001,100200-1
065: 20 20 200001,100200-1
066: 20 20 200001,100200-1
067: 20 20 200001,100200-1
068: 20 20 200001,100200-1
069: 20 20 200001,100200-1
070: 20 20 200001,100200-1
071: 20 20 200001,100200-1
072: 20 20 200001,100200-1
073: 20 20 200001,100200-1
074: 20 20 200001,100200-1
075: 20 20 200001,100200-1
076: 20 20 200001,100200-1
077: 20 20 200001,100200-1
078: 20 20 200001,100200-1
079: 20 20 200001,100200-1
080: 20 20 200001,100200-1
081: 20 20 200001,100200-1
082: 20 20 200001,100200-1
083: 20 20 200001,100200-1
084: 20 20 200001,100200-1
085: 20 20 200001,100200-1
086: 20 20 200001,100200-1
087: 20 20 200001,100200-1
088: 20 20 200001,100200-1
089: 20 20 200001,100200-1
090: 20 20 200001,100200-1
091: 20 20 200001,100200-1
092: 20 20 200001,100200-1
093: 20 20 200001,100200-1
094: 20 20 200001,100200-1
095: 20 20 200001,100200-1
096: 20 20 200001,100200-1
097: 20 20 200001,100200-1
098: 20 20 200001,100200-1
099: 20 20 200001,100200-1
100: 20 20 200001,100200-1
101: 20 20 200001,100200-1
102: 20 20 200001,100200-1
103: 20 20 200001,100200-1
104: 20 20 200001,100200-1
105: 20 20 200001,100200-1
106: 20 20 200001,100200-1
107: 20 20 200001,100200-1
108: 20 20 200001,100200-1
109: 20 20 200001,100200-1
110: 20 20 200001,100200-1
111: 20 20 200001,100200-1
112: 20 20 200001,100200-1
113: 20 20 200001,100200-1
114: 20 20 200001,100200-1
115: 20 20 200001,100200-1
116: 20 20 200001,100200-1
117: 20 20 200001,100200-1
118: 20 20 200001,100200-1
119: 20 20 200001,100200-1
120: 20 20 200001,100200-1
121: 20 20 200001,100200-1
122: 20 20 200001,100200-1
123: 20 20 200001,100200-1
124: 20 20 200001,100200-1
125: 20 20 200001,100200-1
126: 20 20 200001,100200-1
127: 20 20 200001,100200-1
128: 20 20 200001,100200-1
129: 20 20 200001,100200-1
130: 20 20 200001,100200-1
131: 20 20 200001,100200-1
132: 20 20 200001,100200-1
133: 20 20 200001,100200-1
134: 20 20 200001,100200-1
135: 20 20 200001,100200-1
136: 20 20 200001,100200-1
137: 20 20 200001,100200-1
138: 20 20 200001,100200-1
139: 20 20 200001,100200-1
140: 20 20 200001,100200-1
141: 20 20 200001,100200-1
142: 20 20 200001,100200-1
143: 20 20 200001,100200-1
144: 20 20 200001,100200-1
145: 20 20 200001,100200-1
146: 20 20 200001,100200-1
147: 20 20 200001,100200-1
148: 20 20 200001,100200-1
149: 20 20 200001,100200-1
150: 20 20 200001,100200-1
151: 20 20 200001,100200-1
152: 20 20 200001,100200-1
153: 20 20 200001,100200-1
154: 20 20 200001,100200-1
155: 20 20 200001,100200-1
156: 20 20 200001,100200-1
157: 20 20 200001,100200-1
158: 20 20 200001,100200-1
159: 20 20 200001,100200-1
160: 20 20 200001,100200-1
161: 20 20 200001,100200-1
162: 20 20 200001,100200-1
163: 20 20 200001,100200-1
164: 20 20 200001,100200-1
165: 20 20 200001,100200-1
166: 20 20 200001,100200-1
167: 20 20 200001,100200-1
168: 20 20 200001,100200-1
169: 20 20 200001,100200-1
170: 20 20 200001,100200-1
171: 20 20 200001,100200-1
172: 20 20 200001,100200-1
173: 20 20 200001,100200-1
174: 20 20 200001,100200-1
175: 20 20 200001,100200-1
176: 20 20 200001,100200-1
177: 20 20 200001,100200-1
178: 20 20 200001,100200-1
179: 20 20 200001,100200-1
180: 20 20 200001,100200-1
181: 20 20 200001,100200-1
182: 20 20 200001,100200-1
183: 20 20 200001,100200-1
184: 20 20 200001,100200-1
185: 20 20 200001,100200-1
186: 20 20 200001,100200-1
187: 20 20 200001,100200-1
188: 20 20 200001,100200-1
189: 20 20 200001,100200-1
190: 20 20 200001,100200-1
191: 20 20 200001,100200-1
192: 20 20 200001,100200-1
193: 20 20 200001,100200-1
194: 20 20 200001,100200-1
195: 20 20 200001,100200-1
196: 20 20 200001,100200-1
197: 20 20 200001,100200-1
198: 20 20 200001,100200-1
199: 20 20 200001,100200-1
200: 20 20 200001,100200-1
201: 20 20 200001,100200-1
202: 20 20 200001,100200-1
203: 20 20 200001,100200-1
204: 20 20 200001,100200-1
205: 20 20 200001,100200-1
206: 20 20 200001,100200-1
207: 20 20 200001,100200-1
208: 20 20 200001,100200-1
209: 20 20 200001,100200-1
210: 20 20 200001,100200-1
211: 20 20 200001,100200-1
212: 20 20 200001,100200-1
213: 20 20 200001,100200-1
214: 20 20 200001,100200-1
215: 20 20 200001,100200-1
216: 20 20 200001,100200-1
217: 20 20 200001,100200-1
218: 20 20 200001,100200-1
219: 20 20 200001,100200-1
220: 20 20 200001,100200-1
221: 20 20 200001,100200-1
222: 20 20 200001,100200-1
223: 20 20 200001,100200-1
224: 20 20 200001,100200-1
225: 20 20 200001,100200-1
226: 20 20 200001,100200-1
227: 20 20 200001,100200-1
228: 20 20 200001,100200-1
229: 20 20 200001,100200-1
230: 20 20 200001,100200-1
231: 20 20 200001,100200-1
232: 20 20 200001,100200-1
233: 20 20 200001,100200-1
234: 20 20 200001,100200-1
235: 20 20 200001,100200-1
236: 20 20 200001,100200-1
237: 20 20 200001,100200-1
238: 20 20 200001,100200-1
239: 20 20 200001,100200-1
240: 20 20 200001,100200-1
241: 20 20 200001,100200-1
242: 20 20 200001,100200-1
243: 20 20 200001,100200-1
244: 20 20 200001,100200-1
245: 20 20 200001,100200-1
246: 20 20 200001,100200-1
247: 20 20 200001,100200-1
248: 20 20 200001,100200-1
249: 20 20 200001,100200-1
250: 20 20 200001,100200-1
251: 20 20 200001,100200-1
252: 20 20 200001,100200-1
253: 20 20 200001,100200-1
254: 20 20 200001,100200-1
255: 20 20 200001,100200-1
256: 20 20 200001,100200-1
257: 20 20 200001,100200-1
258: 20 20 200001,100200-1
259: 20 20 200001,100200-1
260: 20 20 200001,100200-1
261: 20 20 200001,100200-1
262: 20 20 200001,100200-1
263: 20 20 200001,100200-1
264: 20 20 200001,100200-1
265: 20 20 200001,100200-1
266: 20 20 200001,100200-1
267: 20 20 200001,100200-1
268: 20 20 200001,100200-1
269: 20 20 200001,100200-1
270: 20 20 200001,100200-1
271: 20 20 200001,100200-1
272: 20 20 200001,100200-1
273: 20 20 200001,100200-1
274: 20 20 200001,100200-1
275: 20 20 200001,100200-1
276: 20 20 200001,100200-1
277: 20 20 200001,100200-1
278: 20 20 200001,100200-1
279: 20 20 200001,100200-1
280: 20 20 200001,100200-1
281: 20 20 200001,100200-1
282: 20 20 200001,100200-1
283: 20 20 200001,100200-1
284: 20 20 200001,100200-1
285: 20 20 200001,100200-1
286: 20 20 200001,100200-1
287: 20 20 200001,100200-1
288: 20 20 200001,100200-1
289: 20 20 200001,100200-1
290: 20 20 200001,100200-1
291: 20 20 200001,100200-1
292: 20 20 200001,100200-1
293: 20 20 200001,100200-1
294: 20 20 200001,100200-1
295: 20 20 200001,100200-1
296: 20 20 200001,100200-1
297: 20 20 200001,100200-1
298: 20 20 200001,100200-1
299: 20 20 200001,100200-1
300: 20 20 200001,100200-1
301: 20 20 200001,100200-1
302: 20 20 200001,100200-1
303: 20 20 200001,100200-1
304: 20 20 200001,100200-1
305: 20 20 200001,100200-1
306: 20 20 200001,100200-1
307: 20 20 200001,100200-1
308: 20 20 200001,100200-1
309: 20 20 200001,100200-1
310: 20 20 200001,100200-1
311: 20 20 200001,100200-1
312: 20 20 200001,100200-1
313: 20 20 200001,100200-1
314: 20 20 200001,100200-1
315: 20 20 200001,100200-1
316: 20 20 200001,100200-1
317: 20 20 200001,100200-1
318: 20 20 200001,100200-1
319: 20 20 200001,100200-1
320: 20 20 200001,100200-1
321: 20 20 200001,100200-1
322: 20 20 200001,100200-1
323: 20 20 200001,100200-1
324: 20 20 200001,100200-1
325: 20 20 200001,100200-1
326: 20 20 200001,100200-1
327: 20 20 200001,100200-1
328: 20 20 200001,100200-1
329: 20 20 200001,100200-1
330: 20 20 200001,100200-1
331: 20 20 200001,100200-1
332: 20 20 200001,100200-1
333: 20 20 200001,100200-1
334: 20 20 200001,100200-1
335: 20 20 200001,100200-1
336: 20 20 200001,100200-1
337: 20 20 200001,100200-1
338: 20 20 200001,100200-1
339: 20 20 200001,100200-1
340: 20 20 200001,100200-1
341: 20 20 200001,100200-1
342: 20 20 200001,100200-1
343: 20 20 200001,100200-1
344: 20 20 200001,100200-1
345: 20 20 200001,100200-1
346: 20 20 200001,100200-1
347: 20 20 200001,100200-1
348: 20 20 200001,100200-1
349: 20 20 200001,100200-1
350: 20 20 200001,100200-1
351: 20 20 200001,100200-1
352: 20 20 200001,100200-1
353: 20 20 200001,100200-1
354: 20 20 200001,100200-1
355: 20 20 200001,100200-1
356: 20 20 200001,100200-1
357: 20 20 200001,100200-1
358: 20 20 200001,100200-1
359: 20 20 200001,100200-1
360: 20 20 200001,100200-1
361: 20 20 200001,100200-1
362: 20 20 200001,100200-1
363: 20 20 200001,100200-1
364: 20 20 200001,100200-1
365: 20 20 200001,100200-1
366: 20 20 200001,100200-1
367: 20 20 200001,100200-1
368: 20 20 200001,100200-1
369: 20 20 200001,100200-1
370: 20 20 200001,100200-1
371: 20 20 200001,100200-1
372: 20 20 200001,100200-1
373: 20 20 200001,100200-1
374: 20 20 200001,100200-1
375: 20 20 200001,100200-1
376: 20 20 200001,100200-1
377: 20 20 200001,100200-1
378: 20 20 200001,100200-1
379: 20 20 200001,100200-1
380: 20 20 200001,100200-1
381: 20 20 200001,100200-1
382: 20 20 200001,100200-1
383: 20 20 200001,100200-1
384: 20 20 200001,100200-1
385: 20 20 200001,100200-1
386: 20 20 200001,100200-1
387: 20 20 200001,100200-1
388: 20 20 200001,100200-1
389: 20 20 200001,100200-1
390: 20 20 200001,100200-1
391: 20 20 200001,100200-1
392: 20 20 200001,100200-1
393: 20 20 200001,100200-1
394: 20 20 200001,100200-1
395: 20 20 200001,100200-1
396: 20 20 200001,100200-1
397: 20 20 200001,100200-1
398: 20 20 200001,100200-1
399: 20 20 200001,100200-1
400: 20 20 200001,100200-1
401: 20 20 200001,100200-1
402: 20 20 200001,100200-1
403: 20 20 200001,100200-1
404: 20 20 200001,100200-1
405: 20 20 2
```



# Software for Sale

*If you think that one of our programs looks very interesting, but you can't afford the time to type it in, then our software service will help you out.*

**I**t's three o'clock in the morning. You sit at the computer keyboard having just finished a marathon typing session entering one of the superb programs from *Your Computer*. Your fingers search the keyboard and press the letters R, U and N. You press RETURN, sit back and nothing happens.

Everyone has probably faced this problem. What it does happen is it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little bug slips through unnoticed.

The *Your Computer Software Service* makes available all of the programs from each issue on both cassette and disk at a price of £6.00 for disk and £4.00 for cassette. None of the documentation for the programs is supplied with the software since it is all available in the relevant magazine. Should you not have the magazine then back issues are available from the following address:

INFONET LTD, 3 River Park Estate, Beckenham, Kent, BR4 1HL.  
Tel: (081) 271 7666.

Please contact this address for prices and availability.

## The Disk

Programs on the disk will also be supplied as totally working versions, i.e. where possible we will not use Basic Loaders thus making use of the programs much easier. Unfortunately at the moment we cannot duplicate C16 and Plus 4 cassettes. However programs for these machines will be available on the disk.

What programs are available?

At the top of each article you will find a strip containing the article type, C64 Program etc. So that you can see which programs are available on which format, you will also find a couple of symbols after this strip. The symbols have the following meaning:



This symbol means that the program is available on cassette.



These programs are available on disk.

## Please Note

Since the programs supplied on cassette are total working versions of the programs, we do not just disk-only programs on tape. There is no sense in placing a program that expects to be reading from disk on to tape.

## JANUARY 1989

**PREFAB SPRITES** - A powerful sprite editor for the C64.  
**DATAPOD** - A simple but helpful text processor for the C64. Available on disk and cassette but will only work file on tape.

**UDC COMPRESSOR** - Save on memory when using UDC's in your programs. For C64 only.

**WILLIAM TELL** - Our popular arcade game for the C64.  
**4-AUTOKUN** - Impressive tape loading on your Plus 4 cassette. Only available on disk.

**MINIBASE** - A database for C128 owners.

## ORDER CODE

DISK YJ14ANP11208

TAPE YC26NBB9 (£4.00)

## FEBRUARY 1989

**TAPE MENU** - Add a menu system to your programs cassette (C64).

**SOUND EFFECTS** - A superb sound editor for the C64.  
**F-DUMP** - Dump your C64 text screens to printer with ease.

**DATA LOADER** - A simple way to enter those loads of C64 DATA files.

**SPRITE LIBRARY** - A collection of birds to your growing library (C64).

**PLAY THE GAME** - A superb track machine program for the Plus 4. (Available on disk only).

## ORDER CODE

DISK YJ14FB9 (£6.00)

TAPE YC26NBB9 (£4.00)

MARCH 1989

**PERSONAL FILE** - A cross between a wordprocessor and a database that allows you to set up "cards" that can be quickly altered (C64 disk only).

**LETTER WRITER** - An 80 column text editor for writing those personal letters (C64).

**BASIC WORKSHOP** - A single key entry system, just like a Speccy! (C64).

**HEAD FOR HOME** - One version of a popular board game for C16 and Plus 4 owners - available on disk only.

**SPRITE LIBRARY** - Geometric shapes from this month's instalment (C64).

**ELECTRONIC JOURNAL** - A personal diary on disk (C64 disk only).

**WILLIAM TELL** - Our very popular arcade adventure for the C64.

#### ORDER CODE

DISK YDM1MRS £14.99  
TAPE YCM1MRS £14.99

APRIL 1989

**BASER** - Give your C64 new sound, graphics and toolkit commands as well as a machine code assembler with this BASIC extension.

**AUTOSCROLL** - Professional text screen scrolling with this C64 utility.

**BALANCE SHEET** - Keep your bank manager happy by keeping better track of your money with this C64 program (disk only).

**ICON 64** - Add icon control to your C64 programs with this C64 utility.

**80 COLUMNS** - A suite of 80-col graphic commands for C128 80-column users.

**RELM** - A disk file reader for Plus 4 users. Available on disk only.

**MULTI-PRECISION ARITHMETIC** - A suite of powerful mathematical routines expanding your C64's number crunching abilities.

**STORAGE SPACE** - Use the RAM behind the C64's Kernel and Basic ROMs for data storage (available on disk only).

#### ORDER CODE

DISK YDAPR9E £9.99  
TAPE YCAPR9E £9.99

MAY 1989

**ANTI-FREEZE** - Protect your C64 programs from cartridge-based viruses.

**MULTI-COLOURED LIST** - Brighten up your C64 Basic listings by adding more vibrant colours.

**PRINTEX** - A suite of powerful screen printing commands to add to your Plus 4 Basic (disk only).

**BANKER 128** - Money Management for C128 disk users.

**TURBO TOUCH** - A superb typing trainer for C64 disk users.

#### ORDER CODE

DISK YDSMAY99 £6.99  
TAPE YCSMAY99 £6.99

Commodore disks are available from March 1989. Please ring the editorial office (01-477 0828) for details of these.

#### ORDER FORM - PLEASE COMPLETE IN BLOCK CAPITALS

NAME	QTY	TAPE/DISK	ORDER CODE	PRICE
JUNE 89		TAPE (£4.99)	YCJUNE 89	
JUNE 89		DISK (£6.99)	YDJUNE89	
OVERSEAS POST E1				
			<b>TOTAL</b>	

NAME.....

ADDRESS.....

POSTCODE.....

I enclose a cheque/postal order for £..... made payable to ARGUS SPECIALIST PUBLICATIONS LTD.

All orders should be sent to: COMMODORE READERS SERVICES, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD, HERTS. HP1 1ST

Please allow 28 days for delivery.

# YOUR COMMODORE

Usage: 50p per word.

(including VAT)

Semi-display: £10.00 plus VAT per single column  
continuous minimum 20m. Ring for information on  
series bookings/discounts.

All advertisements in this section must be prepaid.

Advertisements are accepted subject to editorial and commercial  
or word on the relevant payment rate card (available on request).



# 0442 66650

Send your requirements to:  
CLASSIFIED DEPARTMENT  
ASP LTD, ARGUS HOUSE,  
BOUNDARY WAY,  
HEMEL HEMPSTEAD HP2 7ET.

## SOFTWARE



**RAMSHAW COMPUTERS LTD**

Software packages available include:

DATA BASE DESIGN SYSTEMS INC.  
DOS

THE LATEST SOFTWARE FOR  
COMMODORE COMPUTER SYSTEMS

Very few packages are available

to Commodore users at present and

we can help you

**RAMSHAW COMPUTERS LTD**  
14 Station Way, Hemel Hempstead,  
Hertfordshire, HP2 7ET. Tel 0442-666500

## SERVICES

### COMMODORE SUPPORT

Commodore Books	£1.00
Commodore Magazines	£1.00
Commodore Books	£1.00
Commodore Magazines	£1.00
Commodore Books	£1.00
Commodore Magazines	£1.00
Commodore Books	£1.00
Commodore Magazines	£1.00

The above prices

are for Commodore books

and magazines

Please include £1.00 extra for

postage and handling

Order from: Commodore Books

P.O. Box 1000, 10 New Street, West

BIRMINGHAM B3 2EE. Tel 021-555-1234

## HARDWARE

### COMMODORE COMPUTERS

PERSONAL COMPUTER

FOR COMMODORE 64

DATA BASE DESIGN SYSTEMS

DOS

## SOFTWARE

### YOUR COMMODORE SPECIALS

TECH DRAW - A

computer-aided technical

illustration and plotter

program for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

## FOR SALE

### FOR SALE

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore

64/128 and 16/32

systems. Includes

vector drawing, text

editing, scaling, rotation

and plotting, zooming

and much more.

DATA BASE DESIGN

SYSTEMS - A

comprehensive

data base management

system for Commodore



**Readers Problems**

Though the Commodore 64 is one of the world's most popular microcomputers, it can be very difficult to find specific information about your particular machine.

At the Four Comms office we receive literally hundreds of letters from you, our readers, on a wide range of subjects ranging from the simple "Can you give me the telephone number for..." to the more complex "I'm trying to write a program that uses a split screen, how do I do it?"

Unfortunately, the volume of mail received has become so great that it is impossible to answer every letter and still manage to publish a magazine each month.

For this reason we have felt it necessary to produce a number of guidelines for getting information from us:

- 1) We cannot guarantee to answer every letter sent to the magazine. Should it become apparent that a number of readers are suffering from the same problem, then we will reply to the letter via the Letters page.
- 2) A new helpline has been set up. This will be open for your queries on

Tuesday and Thursday afternoons between 2.00pm and 4.00pm. We will not be able to deal with our telephone queries at any other time. If our technical adviser is not available when you ring, then a message will be taken.

3) If you are having problems with one of our listings, can you please let us know in writing. This will enable us to see if a number of people are having the same problem. When a common problem becomes apparent with a program, then a correction sheet will be issued. Enclose a self-addressed, stamped envelope and we will send you a copy of the correction sheet as soon as it is available.

We are sorry that it has become necessary to impose these rules. However, we are sure that you will agree with us that the more time that we can spend making Four Comms the most informative magazine around, the better.

For programme writers to the  
Program Corrections  
Four Comms  
Augus House,  
Boundary Way,  
Harrow Industrial  
HP2 7AT  
Tel: 0808-66251

**Commodore Where Are You?**

At the Four Comms office we are repeatedly asked for the address and telephone number of Commodore

U.K. Many people, after referring to their computer manuals, believe there to be based in Caversham.

The Commodore plant at Caversham was closed down some time ago. Reproduced here you will find the correct

**CORRECTIONS FOR MARCH '89****BREAD FOR HOME**

Once again those indestructible programs have caused trouble.

The programs as it stands will not run correctly. This is due to the fact that quite a lot of code is missing from the loader program. An amendment is on its way. The update will be published as soon as it reaches our office. We apologise for any inconvenience to our Plus 4 readers.

**CORRECTIONS FOR APRIL '89****BALANCE SHEET**

Unfortunately there appeared a couple of errors in the Balance Sheet program from the April Edition.

- 1) Page 31, right hand column, paragraph 4 should read "LIST74" and not Load "LIST".
- 2) The second line of Listing 1 should be POKE776,131:POKE776,164
- 3) All the REM statements should be taken out of Listing 1.
- 4) Again on page 31, right hand column, paragraph 1. This should read:  
50 STX778127@SHEET74

address for Commodore U.K.  
Commodore Business Machines (UK),  
Commodore House, The Swallowbank,  
Berkshire RG10 8XA  
Tel: (0635) 770088  
Quarry Road, Maidenhead,

**The Nobbles By Alan Batchelor**

**AMIGA**

# MICRO Music

**ATARI  
ST**

**IBM PC**

**APPLE  
MAC**

## MUSIC X - The Review

### MICRO MUSIC

*Find out about the magazine  
the computer music world has been  
waiting for*

**KCS - Just what the  
Doctor ordered!!**

**ACORN**

**NEON  
64**

**AMSTRAD  
 CPC**

## THE TIME HAS COME . . .

Britain's First Music Magazine for the Computer User!

Featuring Reviews on Hardware & Software across all formats - Micro Music is

the magazine the Computer Musician has been waiting for.

Available 10th March from all good Newsagents.

**ZX  
SPECTRUM**

# Emlyn Hughes



## INTERNATIONAL FOOTBALL



Quite simply the best football simulation available for the 64. Nothing short of superb.

ZZAP 64

**AVAILABLE NOW FOR  
COMMODORE 64  
£9.95 tape £12.95 disk**  
Coming soon for Spectrum & Amstrad

**GAME  
OF THE YEAR!**  
WINNER OF CCI OSKARS FOR  
BEST SPORTS SIMULATION  
AND BEST GAME OF THE YEAR!

# Audiogenic

Audiogenic Software Limited, Winchester House,  
Canning Road, Harrow HA3 7SJ, England

Order by phone on 01 841 1868

